

**SENSITIVITY ANALYSIS AND APPROXIMATION METHODS FOR GENERAL
EIGENVALUE PROBLEMS**

by

Durbha V. Murthy

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Aerospace Engineering

APPROVED:

Raphael T. Haftka, Chairman

Henry J. Kelley

William L. Hallauer, Jr.

Karl E. Sundkvist

Rakesh K. Kapania

August, 1986

Blacksburg, Virginia

**SENSITIVITY ANALYSIS AND APPROXIMATION METHODS FOR GENERAL
EIGENVALUE PROBLEMS**

by

Durbha V. Murthy

Raphael T. Haftka, Chairman

Aerospace Engineering

(ABSTRACT)

Optimization of dynamic systems involving complex non-hermitian matrices is often computationally expensive. Major contributors to the computational expense are the sensitivity analysis and reanalysis of a modified design. The present work seeks to alleviate this computational burden by identifying efficient sensitivity analysis and approximate reanalysis methods.

For the algebraic eigenvalue problem involving non-hermitian matrices, algorithms for sensitivity analysis and approximate reanalysis are classified, compared and evaluated for efficiency and accuracy. Proper eigenvector normalization is discussed. An improved method for calculating derivatives of eigenvectors is proposed based on a more rational normalization condition and taking advantage of matrix sparsity. Important numerical aspects of this method are also discussed.

To alleviate the problem of reanalysis, various approximation methods for eigenvalues are proposed and evaluated. Linear and quadratic approximations are based directly on the Taylor series. Several

approximation methods are developed based on the generalized Rayleigh quotient for the eigenvalue problem. Approximation methods based on trace theorem give high accuracy without needing any derivatives. Operation counts for the computation of the approximations are given. General recommendations are made for the selection of appropriate approximation technique as a function of the matrix size, number of design variables, number of eigenvalues of interest and the number of design points at which approximation is sought.

Acknowledgements

I wish to express my sincere gratitude to Professor Raphael T. Haftka, for his patience and constant guidance, encouragement and support throughout the course of this research. His advice and technical contributions have proven to be essential in the preparation of this thesis.

I am indebted to Professor Henry J. Kelley, Professor William L. Hallauer, Jr., Professor Karl E. Sundkvist and Professor Rakesh K. Kapania for their helpful suggestions and comments and for serving on the committee. Numerous illuminating discussions with Dr. C. A. Beattie of the Department of Mathematics are gratefully acknowledged. I am also thankful to Dr. K. R. V. Kaza and Dr. R. E. Kielb of NASA Lewis Research Center for their help, interest and essential support.

This research has been made possible through research grant from National Aeronautics and Space Administration (NASA) through grant nos. NAG-3-347 and NAG-1-224. I gratefully acknowledge this financial support.

It is also a pleasure to acknowledge the stimulating environment provided by fellow graduate students at Femoyer Lounge. I am grateful to them and several others for practical assistance given at various times.

Table of Contents

Introduction	1
1.1 Eigenvalue and Eigenvector Derivatives and their Applications	1
1.2 Importance of higher order derivatives	3
1.3 General Matrices	4
1.4 Approximate Updates to Eigenvalues	5
1.5 Objectives of the Present Work	6
1.6 Outline	7
Derivatives of Eigenvalues and Eigenvectors for General Matrices	9
2.1 Problem Definition	9
2.2 Normalization of eigenvectors	11
2.3 Methods of Calculation	14
2.3.1 Adjoint Methods	14
2.3.2 Direct Methods	21
2.3.3 Modification of Direct Method for Banded Matrices	26
2.3.4 Iterative Methods	30

Efficiency Considerations in Calculating Derivatives	31
3.1 Operation Counts	32
3.2 CPU Time Statistics	35
3.3 Calculation of First derivatives of Eigenvalues only	38
3.4 Calculation of First derivatives of Eigenvalues and Eigenvectors	39
3.5 Calculation of First and Second derivatives of Eigenvalues only	42
Approximate Eigenvalues of Modified Systems	45
4.1 Introduction	45
4.2 Derivative Based Approximations	48
4.2.1 Linear Approximation(LIN)	48
4.2.2 Quadratic Approximation(QUAD)	49
4.2.3 Conservative Approximation	50
4.2.4 Generalized Inverse Power Approximation	51
4.2.5 Generalized Hybrid Approximation	52
4.2.6 Reduction Method(RDN)	53
4.3 Rayleigh Quotient Based Approximations	54
4.3.1 Rayleigh Quotient with Nominal Eigenvectors(RAL1)	55
4.3.2 Rayleigh Quotient with Linearly Approximated Eigenvectors(RAL2)	56
4.3.3 Rayleigh Quotient with Perturbed Eigenvectors(RAL3)	56
4.3.4 Rayleigh Quotient with One-step Inverse Iteration(RAL4)	60
4.4 Trace-Theorem Based Approximations	61
4.4.1 One-step Newton-Raphson Iteration(NRT1)	62
4.4.2 Refined One-step Newton-Raphson Iteration(NRT2)	62
4.4.3 One-step Laguerre Iteration(LIT)	64
4.5 Other Approximations	65
4.5.1 [1,1] Pade Approximation(PAD1)	65

Accuracy and Efficiency of Eigenvalue Approximations	68
5.1 Introduction	68
5.2 Accuracy Considerations in Approximating Eigenvalues	69
5.2.1 Order of an Approximation	70
5.2.2 First Order Approximations	70
5.2.3 Second Order Approximations	72
5.2.4 Third Order Approximations	73
5.2.5 Higher Order Approximations	77
5.2.6 Validation of the Theoretical Results	82
5.3 Efficiency Considerations in Approximating Eigenvalues	93
5.4 Discussion of Approximations	96
5.4.1 Case When No Derivatives are Available	96
5.4.2 Case When Derivatives are Available	99
Conclusions	104
References	108
Vita	117

List of Illustrations

Figure 1.	CPU Times for calculation of first derivatives of eigenvalues and eigenvectors for a 60 x 60 matrix	40
Figure 2.	CPU Times for calculation of second derivatives of eigenvalues for a 60 x 60 matrix	43
Figure 3.	Derivation of [1,1] Pade Approximation using Geometric Construction	67
Figure 4.	Relative Percentage errors in the Absolute value of the Eigenvalue 5 x 5 Matrix	83
Figure 5.	Relative Percentage errors in the Real Part of the Eigenvalue 5 x 5 Matrix	84
Figure 6.	Relative Percentage errors in the Imaginary Part of the Eigenvalue 5 x 5 Matrix	85
Figure 7.	Relative Percentage errors in the Absolute value of the Eigenvalue 40 x 40 Matrix	87
Figure 8.	Relative Percentage errors in the Real Part of the Eigenvalue 40 x 40 Matrix	88
Figure 9.	Relative Percentage errors in the Imaginary Part of the Eigenvalue 40 x 40 Matrix	89
Figure 10.	Relative Percentage errors in the Absolute value of the Flutter Eigenvalue 40 x 40 Matrix	90
Figure 11.	Relative Percentage errors in the Real Part of the Flutter Eigenvalue 40 x 40 Matrix	91

Figure 12. Relative Percentage errors in the Imaginary Part of the Flutter
Eigenvalue 40 x 40 Matrix 92

List of Tables

Table 1. Operation Counts	34
Table 2. Correlation between Operation Count(OC) and CPU time	36
Table 3. Correlation between Operation Count(OC) and CPU time(Contd.)	37
Table 4. Operation Counts for First and Second Order Approximations	94
Table 5. Operation Counts for Third and Higher Order Approximations	95
Table 6. Operation Counts for First Order Approximations when Eigenvalue Derivatives are Free	100
Table 7. Operation Counts for First and Second Order Approximations when All First Derivatives are Free	102
Table 8. Operation Counts for Third and Higher Order Approximations when All First Derivatives are Free	103

List of Symbols

a	Subscript indicating approximate value
a_1	Subscript indicating a first approximation
A	Complex general system matrix of order n
\mathbf{a}_m^T	m -th row of A with the m -th column deleted
B	$[A - \lambda^{(k)}I \mid -\mathbf{u}^{(k)}]_{m\text{-th column deleted}}$
\mathbf{b}_m	$[C^T]^{-1}\mathbf{a}_m$
c	Arbitrary constant
$c_{kj\alpha}$	Contribution of the j -th right eigenvector to $\mathbf{u}_\alpha^{(k)}$
C	$(A - \lambda I)_{m\text{-th row and column deleted}}$
$d_{kj\alpha}$	Contribution of the j -th left eigenvector to $\mathbf{v}_\alpha^{(k)}$
e_{kj}	Contribution of the j -th right eigenvector to $\delta\mathbf{u}^{(k)}$
f_{kj}	Contribution of the j -th left eigenvector to $\delta\mathbf{v}^{(k)}$
$f(\lambda)$	Ratio of $p'(\lambda)$ and $p(\lambda)$
I	Identity matrix of order n
i	Index of eigenvalue or eigenvector
j	Index of eigenvalue or eigenvector

k	Index of eigenvalue or eigenvector
l	Number of eigenvalues of interest
m	Number of design variables
$\binom{m}{2}$	Number of combinations of m objects taken two at a time
n	Order of matrices A and A_0
p	Vector of design variables
p_α	α -th design variable
$p(\lambda)$	Characteristic equation of A
P	Matrix of order 2 defined in eq. (4.2.9)
Q	Matrix of order 2 defined in eq. (4.2.9)
r	Right hand side in Direct method
R	Generalized Rayleigh Quotient
s	Order of Approximation
s	Right hand side in Direct method
t	t_{m -th row deleted
trace	Sum of the elements on the principal diagonal
T	Superscript to denote the transpose
$u^{(k)}$	k -th right eigenvector of matrix A
$u_m^{(k)}$	m -th element of $u^{(k)}$
U	Matrix whose columns are the right eigenvectors
$v^{(k)}$	k -th left eigenvector of matrix A
$v_m^{(k)}$	m -th element of $v^{(k)}$
V	Matrix whose columns are the left eigenvectors

\mathbf{x}	$\mathbf{u}_{m\text{-th row deleted}}$
\mathbf{y}	$\mathbf{v}_{m\text{-th row deleted}}$
\mathbf{y}_1	Defined in eq. (2.3.28)
\mathbf{y}_2	Defined in eq. (2.3.32)
\mathbf{Z}	Defined in eq. (5.2.11)
α	Index of design variable
$, \alpha$	Subscript indicating derivative with respect to p_α
$, \alpha\beta$	Subscript indicating second derivative with respect to p_α and p_β
$, \beta$	Subscript indicating derivative with respect to p_β
δ	Prefix indicating estimated change
Δ	Prefix indicating actual change
ζ_α	Value of design variable lying between p_α and $p_\alpha + \Delta p_\alpha$
κ	Measure of the sparsity of \mathbf{A}_α or $\Delta\mathbf{A}$
λ	Eigenvalue of \mathbf{A}
λ_0	Eigenvalue of \mathbf{A}_0
Λ	Diagonal matrix of eigenvalues
*	Superscript to denote the conjugate transpose
0	Subscript indicating nominal value

Chapter 1

Introduction

1.1 Eigenvalue and Eigenvector Derivatives and their Applications

Dynamic response and loads are an important consideration in the understanding and design of many physical systems. The analytical models for a wide range of these systems are governed by linear differential equations so that dynamic model analysis often consists of the solution of an eigenvalue problem. The eigenvalues and the eigenvectors of the system are fundamental quantities employed in determining the behavior of the system. Variations in system parameters lead to changes in the eigenvalues and the eigenvectors and hence in the response characteristics of the system. It is important to

know the magnitude of these variations, and this information is contained in the derivatives of the system eigenvalues and eigenvectors. Thus derivatives of eigenvalues and eigenvectors are of immense interest in several fields of physical sciences and engineering and much research effort has been expended in developing methods to calculate them.

The applications of these derivatives (or synonymously, sensitivities) are manifold. Probably the most important applications are in the area of design optimization. System response sensitivities provide vital information in an optimization procedure and in general the cost of calculating derivatives is the dominant contributor to the total cost in an optimization procedure so that the efficient computation of eigenvalue and eigenvector derivatives is desirable. Derivatives can also be effectively used to approximate the eigenvalues and eigenvectors of a modified system and thus reduce the cost of reanalysis, substantially lowering the computational burden in optimization tasks. The derivatives are very useful even in non-automated design procedures because it is often not clear, from analysis alone, how to modify a design to improve or maintain the desirable properties. The derivatives identify design parameters that have the most or the least influence on the design process and thus ease the effort in design trend studies.

Derivatives of eigenvalues and eigenvectors are particularly valuable in calculating the statistics of eigenvalue locations in stochastic dynamic systems. All physical systems are essentially subject to random environments and the effect of randomly changing environments is crucial for such systems

as missiles, spacecraft, airplanes, land vehicles, buildings and machinery. In addition, many system models do not have well-defined properties and it is frequently difficult to predict these properties (for example, stiffnesses) accurately[1-4]. The uncertainties in the system eigenvalues and eigenvectors are calculated from the estimated uncertainties in the properties of the system and the environment by using the derivatives of eigenvalues and eigenvectors.

The application of derivatives is not restricted to design-oriented activities. Sensitivity analysis is also playing an increasing role in determining the analytical model itself. In the areas of system identification and analytical model improvement using test results, sensitivity analysis is of growing importance. Much recent work in these fields is directly dependent on the calculation of eigensystem derivatives.

1.2 Importance of higher order derivatives

While in the past attention was mostly restricted to first order derivatives of eigenvalues, higher order derivatives are assuming a greater importance recently. It has been found in certain cases that second order derivatives are very effective in improving accuracy of approximations[5-13] and efficiency of design[7,11,12]. In almost all instances, eigenvalues are non-linear functions of system parameters and a second order approximation offers a wider range of applicability compared to the first order approximation. Intermediate

variables which may improve the quality of first order approximations are not generally available for eigenvalue approximations. Also, some optimization algorithms require second order derivatives, and first order derivatives of optimal solutions require second order derivatives of constraints[13]. The use of second derivatives can also greatly reduce the number of reanalyses required for the convergence of an optimization procedure[11,14]. Further, in certain optimization algorithms, second order approximations for eigenvalue constraints can drastically relax the move limits, thus achieving a nearly optimum trajectory, and can virtually eliminate the need for trial and error adjustment of move limits, thus improving the performance of the optimizer[14]. Looking at another aspect, in problems where instabilities are to be avoided, a first order calculation may completely fail to detect instabilities[6]. References [15,16] also offer examples of the usefulness of second order derivatives.

1.3 General Matrices

The problem of calculating the derivatives of symmetric and hermitian eigenproblems is relatively simple and solution procedures are well-established, e.g.[17-21]. However, many physical problems give rise to non-self-adjoint formulations and thus lead to general matrices. An important example is aeroelastic stability which requires the solution of eigenproblems

with complex, general and fully populated matrices. General matrices are also obtained in damped structural systems and in network analysis and control system design where the eigenvalues are usually called poles. In the present study, the emphasis is on general matrices and the special properties of matrices, such as symmetry, are not considered.

1.4 Approximate Updates to Eigenvalues

Eigenvalue calculation for any but the smallest systems is an expensive process and is a major contributor to the computational expense of the typical dynamic analysis. In the design of structural systems, an iterative design/analysis process is performed until a satisfactory design is achieved. The cycle of the iterative process consists of an update of the structural design, a response analysis and calculation of updated responses and loads. When the process is not automated, a revision of the mathematical model may also be present. In typical dynamic analyses, each design cycle is an expensive process because the mathematical models are large and the time and effort required for analysis of an updated design are often prohibitive and greatly reduce the effectiveness of the design process. Apart from the computational effort, organizational effort can also be substantial.

Appreciable cost savings can be realized if a quick evaluation of a change in system response resulting from the design changes is possible. An

efficient, even if approximate, evaluation of eigenvalues of a modified system is valuable in these applications. The value of an approximation depends on its efficiency as well as its accuracy in applications. A quick approximation that is valid in a very limited range of design space is of little use as it can severely reduce the global efficiency by requiring many more evaluations for convergence of the design process.

Several researchers have worked on suitable approximations for eigenvalues of a modified design. However, in the past, attention seems to have been restricted to real symmetric systems which have eigenvalues in the real number field. It is one of the objectives of the present work to extend the techniques of approximation to general (complex non-hermitian) systems and perform a comparative analysis of the various techniques.

1.5 Objectives of the Present Work

The objectives of the present research are to:

1. Review and perform a comparative analysis of the various methods available for calculating the derivatives of eigenvalues and eigenvectors of general matrices.
2. Propose and evaluate some modifications to existing techniques.

3. Formulate guidelines for selecting the most efficient computational algorithm for particular applications.
4. Review the various approximations to eigenvalues for real symmetric systems and extend them to the case of complex general systems.
5. Compare the various approximations in terms of efficiency and accuracy for some systems.

1.6 Outline

Chapter 2 reviews the various methods available for the calculation of derivatives of eigenvalues and eigenvectors for general matrices. The important consideration of normalization of eigenvectors of complex general matrices, which has not been adequately dealt with in the literature, is discussed. A new algorithm to calculate the derivatives of eigenvalues and eigenvectors simultaneously, based on a better normalizing condition, is described and important numerical aspects regarding the implementation of the algorithm are discussed, with consideration being given to sparse matrices. The various algorithms are classified as Adjoint or Direct.

The efficiency considerations of the various algorithms are examined in Chapter 3. Operation counts are presented in terms of matrix size, number of design parameters, and the number of eigenvalues and eigenvectors of

interest. Actual CPU times are also presented for typical matrices for a range of parameters that influence the efficiency of the algorithms.

Chapter 4 provides a survey of approximation methods proposed in the literature for real symmetric matrices, describing their special features. The approximation methods are extended to complex general matrices wherever feasible. Some approximation methods which do not seem to have been applied in the past are also presented. The approximation methods are classified on the basis of their theoretical origin.

Numerical results from applying the proposed techniques of approximations are presented in Chapter 5. Operation counts are presented in terms of matrix size, number of design parameters, number of eigenvalues of interest and the number of times the approximation is to be performed. The approximation techniques are applied to typical matrices and random matrices and are evaluated in terms of their accuracy and efficiency.

Chapter 6 contains the concluding remarks. General guidelines for selecting approximation methods and algorithms for calculation of eigenvalue and eigenvector derivatives are summarized. This chapter also contains remarks about the limitations of the present work and recommendations for further research.

Chapter 2

Derivatives of Eigenvalues and Eigenvectors for General Matrices

2.1 Problem Definition

The matrix eigenproblem is defined as follows:

$$\mathbf{A}\mathbf{u}^{(k)} = \lambda^{(k)}\mathbf{u}^{(k)} \quad (2.1.1)$$

and the corresponding adjoint problem is

$$\mathbf{v}^{(k)T}\mathbf{A} = \lambda^{(k)}\mathbf{v}^{(k)T} \quad (2.1.2)$$

where \mathbf{A} is a general complex matrix of order n and $\lambda^{(k)}$, $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ are the k -th eigenvalue and right and left eigenvectors respectively. A superscript T denotes the transpose.

(The adjoint problem is defined by some authors in an alternative form as

$$\mathbf{v}^{(k)*} \mathbf{A} = \lambda^{(k)} \mathbf{v}^{(k)*}$$

where superscript $*$ denotes a conjugate-transpose. However, the notation of eq. (2.1.2) is more popular in the literature on structural dynamics).

The eigenvalues and eigenvectors are complex and do not necessarily occur in complex-conjugate pairs. All eigenvalues are assumed to be distinct.

The matrix \mathbf{A} and hence, $\lambda^{(k)}$, $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ are functions of design parameter vector \mathbf{p} with individual parameters denoted by Greek subscripts, e.g. p_α . Derivatives with respect to p_α are denoted by the subscript, α e.g., $\frac{\partial \mathbf{A}}{\partial p_\alpha} = \mathbf{A}_{,\alpha}$. All the design variables are assumed to be real.

The well-known biorthogonality properties of the eigenvectors are given by

$$\mathbf{v}^{(i)T} \mathbf{u}^{(j)} = 0 \text{ iff } i \neq j \quad (2.1.3)$$

and

$$\mathbf{v}^{(i)T} \mathbf{A} \mathbf{u}^{(j)} = 0 \text{ iff } i \neq j \quad (2.1.4)$$

Note that, the left hand side of eq. (2.1.3) is not an inner product as usually understood, since $\mathbf{v}^{(l)}$ and/or $\mathbf{u}^{(l)}$ may be complex vectors. Note also that the left eigenvectors of \mathbf{A} are the right eigenvectors of \mathbf{A}^T and vice versa. \wedge

2.2 Normalization of eigenvectors

The eigenvectors $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ are not completely defined by eqs. (2.1.1) and (2.1.2). A normalization condition has to be imposed to obtain unique eigenvectors. For brevity, let us consider only the normalization of the right eigenvector. A normalizing condition frequently imposed in the self-adjoint case is the following:

$$\mathbf{u}^{(k)T} \mathbf{u}^{(k)} = 1 \quad (2.2.1)$$

However, it is not always possible to use eq. (2.2.1) for non-self-adjoint problems as $\mathbf{u}^{(k)T} \mathbf{u}^{(k)}$ can equal zero or a very small number causing numerical difficulties. This is true even if the matrix \mathbf{A} is real, as shown by the example matrix $\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Unfortunately, considerable confusion exists in the literature regarding this point and several authors arbitrarily adopted eq.(2.2.1) as a normalizing condition for non-self-adjoint problems, e.g.[11,12,15,22-25]. In this respect, the formulations of these references are not rigorous for general matrices.

One possible way to avoid the above difficulty is to replace eq.(2.2.1) by

$$\mathbf{u}^{(k)*} \mathbf{u}^{(k)} = 1 \quad (2.2.2)$$

where superscript * denotes a conjugate-transpose. Eq. (2.2.2) is not prone to the difficulties of eq. (2.2.1) because $\mathbf{u}^{(k)*} \mathbf{u}^{(k)}$ is always guaranteed to be non-zero. But, eq.(2.2.2) is not a complete normalizing condition as it does not render the eigenvector unique. If \mathbf{u} satisfies eq.(2.2.2), then $\mathbf{w} = \mathbf{u}e^{ic}$, where $i = \sqrt{-1}$ and c is an arbitrary real number, also satisfies eq.(2.2.2). Despite this limitation, eq.(2.2.2) can be used satisfactorily in certain formulations[26,27].

Another normalization condition, inspired by the biorthogonality property of the left and right eigenvectors, is

$$\mathbf{v}^{(k)T} \mathbf{u}^{(k)} = 1 \quad (2.2.3)$$

Eq.(2.2.3) also does not render the eigenvectors unique, since a pair $\mathbf{u}^{(k)}, \mathbf{v}^{(k)}$ can be replaced by $c\mathbf{u}^{(k)}, (1/c)\mathbf{v}^{(k)}$, where c is an arbitrary non-zero complex number, and still satisfy eq. (2.2.3). Again, this is not necessarily a severe restriction for calculation of the derivatives of eigenvectors[5,16,28-30]. It must, however, be emphasized that if the eigenvector is not unique, nor is its derivative.

The normalization condition

$$u_m^{(k)} = 1 \quad (2.2.4)$$

is very attractive because it renders the eigenvectors unique and at the same time, the index m can be chosen easily to avoid ill-conditioning. Apparently, only Nelson[31] used this normalizing condition in obtaining the derivatives of eigenvectors. The index, m , may be chosen such that

$$|u_m^{(k)}| = \max_j |u_j^{(k)}| \quad (2.2.5)$$

Another choice for m , used by Nelson[31], is

$$|u_m^{(k)}| |v_m^{(k)}| = \max_j |u_j^{(k)}| |v_j^{(k)}| \quad (2.2.6)$$

The nature of uncertainty of the derivative of the eigenvector is of some interest. Without a normalizing condition, an eigenvector is uncertain to the extent of a non-zero constant multiplier. The derivative of an eigenvector is uncertain to the extent of an additive multiple of that eigenvector. To show this, let $\mathbf{u}^{(k)}$ be an eigenvector so that $\mathbf{w}^{(k)} = c\mathbf{u}^{(k)}$ is also an eigenvector. Then, if p_α is a design parameter,

$$\frac{\partial \mathbf{w}^{(k)}}{\partial p_\alpha} = \frac{\partial (c\mathbf{u}^{(k)})}{\partial p_\alpha} = c \frac{\partial \mathbf{u}^{(k)}}{\partial p_\alpha} + d\mathbf{u}^{(k)} \quad (2.2.7)$$

The quantity $d = \frac{\partial c}{\partial p_\alpha}$ is not zero since the quantity c is not really a constant, but is a function of the nature of the normalization criterion. In practice, the constant d depends on the way the eigenvectors $\mathbf{u}^{(k)}$ and $\mathbf{w}^{(k)}$ are normalized.

2.3 Methods of Calculation

The various methods of calculating the derivatives of eigenvalues and eigenvectors can be divided into three categories:

1. *Adjoint Methods*, which use both the right and the left eigenvectors.
2. *Direct Methods*, which use only the right eigenvectors.
3. *Iterative Methods*, which use an iterative algorithm that converges to the required derivatives.

2.3.1 Adjoint Methods

The first expressions for the derivatives of eigenvalues of a general matrix seem to have been derived by Lancaster[32]. Considering only a single parameter, Lancaster obtained the following expressions for the first and second derivatives of an eigenvalue:

$$\lambda_{,\alpha}^{(k)} = \frac{\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)}}{\mathbf{v}^{(k)T} \mathbf{u}^{(k)}} \quad (2.3.1)$$

$$\lambda_{,\alpha\alpha}^{(k)} = \frac{\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha\alpha} \mathbf{u}^{(k)}}{\mathbf{v}^{(k)T} \mathbf{u}^{(k)}} + 2 \sum_{\substack{j=1 \\ j \neq k}}^n \left[\frac{(\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(j)}) (\mathbf{v}^{(j)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)})}{(\lambda^{(k)} - \lambda^{(j)}) (\mathbf{v}^{(k)T} \mathbf{u}^{(k)}) (\mathbf{v}^{(j)T} \mathbf{u}^{(j)})} \right] \quad (2.3.2)$$

Eq. (2.3.1) can be obtained in the following manner. Differentiate eq. (2.1.1) with respect to the parameter p_α to obtain

$$\mathbf{A}_{,\alpha} \mathbf{u}^{(k)} + \mathbf{A} \mathbf{u}_{,\alpha}^{(k)} = \lambda_{,\alpha}^{(k)} \mathbf{u}^{(k)} + \lambda^{(k)} \mathbf{u}_{,\alpha}^{(k)} \quad (2.3.3)$$

Premultiplying both sides by $\mathbf{v}^{(k)T}$, we get

$$\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} + \mathbf{v}^{(k)T} \mathbf{A} \mathbf{u}_{,\alpha}^{(k)} = \mathbf{v}^{(k)T} \lambda_{,\alpha}^{(k)} \mathbf{u}^{(k)} + \mathbf{v}^{(k)T} \lambda^{(k)} \mathbf{u}_{,\alpha}^{(k)} \quad (2.3.4)$$

The last terms in the expressions on both sides are equal due to eq. (2.1.2), so that

$$\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} = \lambda_{,\alpha}^{(k)} \mathbf{v}^{(k)T} \mathbf{u}^{(k)} \quad (2.3.5)$$

Eq. (2.3.1) follows immediately. Eq. (2.3.2) is derived in its more general form later. An expression corresponding to eq. (2.3.1) for a non-linear eigenvalue problem

$$\mathbf{A}(\lambda) \mathbf{u}^{(k)} = \mathbf{0} \quad (2.3.6)$$

was obtained by Pedersen and Seyranian[33] in a similar manner as

$$\lambda_{,\alpha}^{(k)} = \frac{\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)}}{\mathbf{v}^{(k)T} \mathbf{A}_{,\lambda} \mathbf{u}^{(k)}} \quad (2.3.7)$$

To obtain the second derivatives of eigenvalues, the first derivatives of left and right eigenvectors are calculated either explicitly[5,12,16,26,30] as in eq. (2.3.16) or implicitly[12,15,32] as in eq. (2.3.18). Since the eigenvalues are assumed to be distinct, the set of eigenvectors forms a basis for the n -space and the first derivatives of eigenvectors can be expressed in terms of the eigenvectors as

$$\mathbf{u}_{,\alpha}^{(k)} = \sum_{j=1}^n c_{kj\alpha} \mathbf{u}^{(j)} \quad \text{and} \quad \mathbf{v}_{,\alpha}^{(k)} = \sum_{j=1}^n d_{kj\alpha} \mathbf{v}^{(j)} \quad (2.3.8)$$

Now, the calculation of the first derivatives of eigenvectors reduces to the evaluation of the coefficients $c_{kj\alpha}$ and $d_{kj\alpha}$.

Premultiplying eq. (2.3.3) by $\mathbf{v}^{(j)T}$, where $j \neq k$, we get

$$\mathbf{v}^{(j)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} + \mathbf{v}^{(j)T} \mathbf{A} \mathbf{u}_{,\alpha}^{(k)} = \mathbf{v}^{(j)T} \lambda_{,\alpha}^{(k)} \mathbf{u}^{(k)} + \mathbf{v}^{(j)T} \lambda^{(k)} \mathbf{u}_{,\alpha}^{(k)} \quad (2.3.9)$$

Now, substituting the expansions of eq. (2.3.8) and using eqs. (2.1.1) and (2.1.2) and the bi-orthogonality property of eq. (2.1.3), we obtain

$$c_{kj\alpha} = \frac{\mathbf{v}^{(j)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)}}{(\lambda^{(k)} - \lambda^{(j)}) \mathbf{v}^{(j)T} \mathbf{u}^{(j)}} \quad k \neq j \quad (2.3.10)$$

Proceeding in a similar manner after differentiating eq. (2.1.2) with respect to the parameter p_α , we obtain

$$d_{kja} = \frac{\mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(j)}}{(\lambda^{(k)} - \lambda^{(j)}) \mathbf{v}^{(j)T} \mathbf{u}^{(j)}} \quad k \neq j \quad (2.3.11)$$

The above expressions for the coefficients c_{kja} and d_{kja} were obtained by Rogers[29].

It can be observed that

$$d_{kja} = -c_{jka} \frac{\mathbf{v}^{(k)T} \mathbf{u}^{(k)}}{\mathbf{v}^{(j)T} \mathbf{u}^{(j)}} \quad (2.3.12)$$

Reddy[34] derived an equivalent expression for the response derivative by casting the derivative as the solution of a forced response problem for the same system.

Note that, in view of eq. (2.2.7), the coefficients c_{kka} and d_{kka} in eq. (2.3.8) are arbitrary and depend on the normalization of the eigenvectors. For example, if eq. (2.2.4) is used to normalize the right eigenvectors, then

$$c_{kka} = - \sum_{\substack{j=1 \\ j \neq k}}^n c_{kja} u_m^{(j)} \quad (2.3.13)$$

and if eq. (2.2.3) is used to normalized the left eigenvectors, then

$$d_{kka} = -c_{kka} \quad (2.3.14)$$

It has been proposed[31,35,36] that the eigenvector derivative be approximated by using less than the full set of eigenvectors in the expansion of eq. (2.3.8) so that the evaluation of eigenvector derivative by Adjoint method could become cheaper. This variant of Adjoint method has received mixed reports in the literature[31,35]. The quality of such an approximation is difficult to assess beforehand and the selection of the number of eigenvectors to be retained in the expansion is problem dependent. It is not considered in this work because a meaningful comparison with other methods cannot be easily be made. However, this consideration should not be ignored while implementing the sensitivity calculations for particular problems.

The expressions for the second derivatives of eigenvalues were obtained by Plaut and Huseyin[30]. For the sake of simplicity in expressions, let us assume, without loss of generality, that the right and left eigenvectors are normalized as in eq. (2.2.3). Eq. (2.3.1) can then be written as

$$\lambda_{,\alpha}^{(k)} = \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} \quad (2.3.15)$$

Differentiating with respect to a parameter p_β uncorrelated to the parameter p_α , we obtain

$$\lambda_{,\alpha\beta}^{(k)} = \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} + \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}_{,\beta}^{(k)} + \mathbf{v}_{,\beta}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} \quad (2.3.16)$$

which can be equivalently written, without involving the derivative of the left eigenvector, as

$$\lambda_{,\alpha\beta}^{(k)} = \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} + \mathbf{v}^{(k)T} (\mathbf{A}_{,\alpha} - \lambda_{,\alpha} \mathbf{I}) \mathbf{u}_{,\beta}^{(k)} + \mathbf{v}^{(k)T} (\mathbf{A}_{,\beta} - \lambda_{,\beta} \mathbf{I}) \mathbf{u}_{,\alpha}^{(k)} \quad (2.3.17)$$

Eq. (2.3.16) can be rewritten using eqs. (2.3.10) and (2.3.11) as

$$\lambda_{,\alpha\beta}^{(k)} = \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} + \sum_{\substack{j=1 \\ j \neq k}}^n (\lambda^{(k)} - \lambda^{(j)}) (c_{kj\alpha} d_{kj\beta} + c_{kj\beta} d_{kj\alpha}) \quad (2.3.18)$$

Crossley and Porter[5,28] derived similar expressions for derivatives with respect to the elements of the matrix. Expression for the N -th order diagonal derivative was derived by Elrazaz and Sinha[9] and it is

$$\frac{\partial^N \lambda^{(k)}}{\partial p_\alpha^N} = \mathbf{v}^{(k)T} \left[\begin{array}{l} \frac{\partial^N \mathbf{A} \mathbf{u}^{(k)}}{\partial p_\alpha^N} + N \frac{\partial^{N-1} \mathbf{A} \partial \mathbf{u}^{(k)}}{\partial p_\alpha^{N-1} \partial p_\alpha} + \\ \frac{N(N-1)}{2!} \frac{\partial^{N-2} \mathbf{A} \partial^2 \mathbf{u}^{(k)}}{\partial p_\alpha^{N-2} \partial p_\alpha^2} + \\ \dots + N \frac{\partial \mathbf{A} \partial^{N-1} \mathbf{u}^{(k)}}{\partial p_\alpha \partial p_\alpha^{N-1}} - \\ \frac{N(N-1)}{2!} \frac{\partial^{N-2} \lambda^{(k)} \partial^2 \mathbf{u}^{(k)}}{\partial p_\alpha^{N-2} \partial p_\alpha^2} - \\ \dots - N \frac{\partial \lambda^{(k)} \partial^{N-1} \mathbf{u}^{(k)}}{\partial p_\alpha \partial p_\alpha^{N-1}} \end{array} \right] \quad (2.3.19)$$

Morgan[37] developed a different computational approach for the derivative of an eigenvalue without requiring the eigenvectors explicitly. His expression is equivalent to

$$\lambda_{,\alpha}^{(k)} = \frac{\text{trace of } \{ [\text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I})] \mathbf{A}_{,\alpha} \}}{\text{trace of } \text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I})} \quad (2.3.20)$$

The corresponding expression for derivatives with respect to matrix elements was derived by Nicholson[38].

It can however be shown that[39]

$$\text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I}) = t_k \mathbf{u}^{(k)} \mathbf{v}^{(k)T} \quad (2.3.21)$$

where t_k is a constant and that[40]

$$\begin{aligned} \text{trace of } \{ [\text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I})] \mathbf{A}_{,\alpha} \} &= t_k \mathbf{v}^{(k)T} \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} \\ \text{trace of } \text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I}) &= t_k \mathbf{v}^{(k)T} \mathbf{u}^{(k)} \end{aligned} \quad (2.3.22)$$

Thus, in the computation of $\text{adj}(\mathbf{A} - \lambda^{(k)}\mathbf{I})$, both right and left eigenvectors are implicitly computed, in view of eq. (2.3.21). Eqs. (2.3.22) also show that Morgan's eq. (2.3.20) is equivalent to Lancaster's eq. (2.3.1). Woodcock[41] also obtained formulas involving the adjoint matrix for the first and second derivatives of eigenvalues. An operation count shows that calculation of the adjoint matrix is several times more expensive than the explicit calculation of right and left eigenvectors so that Lancaster's formula is preferable to formulas requiring the adjoint matrix. This conclusion is also supported by sample computations[42]. In addition, although eq. (2.3.20) was used satisfactorily for small problems[43,44], numerical difficulties were reported for reasonably large problems[45]. Woodcock's formula for the second derivative

of the eigenvalue requires a partial derivative of the adjoint matrix and this is so complicated that Woodcock himself recommends the finite difference method. Formulas due to Morgan and Woodcock are not therefore considered in the following.

In calculating the derivatives by Adjoint Methods, i.e., using eqs. (2.3.1), (2.3.8)-(2.3.18),

- the first derivative of an eigenvalue requires the corresponding right and left eigenvectors.
- the first derivative of an eigenvector requires *all* the left and right eigenvectors.
- the second derivative of an eigenvalue requires the corresponding right and left eigenvectors and their first derivatives.

2.3.2 Direct Methods

The second category comprises methods that evaluate the derivatives using only the right eigenproblem. Direct methods typically involve either the evaluation of the characteristic polynomial or the solution of a system of linear simultaneous equations without requiring all the left and right eigenvectors. Methods requiring the evaluation of the characteristic polynomial and the derivative of the determinant[45,46] are $O(n^5)$ processes while other methods

considered here are at most $O(n^3)$ processes. In addition, the determination of the characteristic polynomial is, in general, an unsatisfactory process with respect to numerical stability, even when all the eigenvalues are well-conditioned[47]. While numerically stable algorithms have been proposed for evaluation of the characteristic polynomial[48], the computational expense still seems to be formidable. Hence, we do not consider these methods. Methods requiring the solution of a system of equations have the particularly attractive feature that the coefficient matrix needs to be factored only once for each eigenvalue regardless of the number of parameters and the order of the derivatives required. Thus, they are very useful in applications where higher order derivatives are required.

The earliest method in this class is due to Garg[22] who obtained the first derivatives of the eigenvalue and the eigenvector by solving two systems of $(n + 1)$ equations each in the real domain, without requiring any left eigenvectors. However, his formulation involves several matrix multiplications. Rudisill[23] proposed a scheme in which only the corresponding left and right eigenvectors are required to calculate the first derivative of the eigenvalue and the eigenvector. This was refined by Rudisill and Chu[24] to avoid calculating the left eigenvectors altogether. Solution of a system of only $(n + 1)$ equations is required (though in the complex domain) to obtain the first derivatives of eigenvalue as well as eigenvector. Extension to higher order derivatives is straightforward. Cardani and Mantegazza[25]

proposed solution methods of the same formulation for sparse matrices and extended it to the quadratic eigenproblem.

One weakness that is common to all the above formulations that do not require left eigenvectors[22-25] is that they rely on the normalization condition given by eq. (2.2.1), which is unreliable as discussed earlier.

Nelson[31] circumvented this difficulty by using the normalizing conditions

$$\mathbf{v}^{(k)T} \mathbf{u}^{(k)} = 1 \quad \text{and} \quad u_m^{(k)} = 1 \quad (2.3.23)$$

However, the formulation of Rudisill and Chu is superior to Nelson's formulation in that it does not require any left eigenvectors.

In this work, we propose a variation of the Rudisill and Chu formulation which does not rely on the questionable normalizing condition of eq. (2.2.1) and at the same time requires no left eigenvectors.

Differentiating eq. (2.1.1), we get

$$\mathbf{A} \mathbf{u}_{,\alpha}^{(k)} + \mathbf{A}_{,\alpha} \mathbf{u}^{(k)} = \lambda_{,\alpha}^{(k)} \mathbf{u}_{,\alpha}^{(k)} + \lambda_{,\alpha}^{(k)} \mathbf{u}^{(k)} \quad (2.3.24)$$

which can be rewritten in partitioned matrix form as

$$\left[\mathbf{A} - \lambda^{(k)} \mathbf{I}; \quad -\mathbf{u}^{(k)} \right] \left\{ \begin{array}{c} \mathbf{u}_{,\alpha}^{(k)} \\ \lambda_{,\alpha}^{(k)} \end{array} \right\} = -\mathbf{A}_{,\alpha} \mathbf{u}^{(k)} \quad (2.3.25)$$

Now, we impose the normalizing condition of eq. (2.2.4). Differentiation of eq. (2.2.4) yields,

$$u_{m,\alpha}^{(k)} = 0 \quad (2.3.26)$$

Because of eq. (2.3.26), the m -th column of the coefficient matrix in eq. (2.3.25) can be deleted. Eq. (2.3.26) also reduces the number of unknowns by one so that eq. (2.3.25) is now a system of n equations in n unknowns. Eq. (2.3.25) is rewritten as

$$\mathbf{B} \mathbf{y}_1 = \mathbf{r} \quad (2.3.27)$$

where

$$\mathbf{B} = [\mathbf{A} - \lambda^{(k)} \mathbf{I} \quad ; \quad -\mathbf{u}^{(k)}]_{m\text{-th column deleted}}$$

$$\mathbf{y}_1 = \left\{ \begin{array}{c} \mathbf{u}_{,\alpha}^{(k)} \\ -\lambda_{,\alpha}^{(k)} \end{array} \right\} \text{ with } m\text{-th element deleted}$$

$$\mathbf{r} = -\mathbf{A}_{,\alpha} \mathbf{u}^{(k)} \quad (2.3.28)$$

To get second derivatives, differentiate (2.3.24) with respect to p_β and get,

$$\begin{aligned} (\mathbf{A} - \lambda^{(k)} \mathbf{I})_{,\alpha\beta} \mathbf{u}_{,\alpha\beta}^{(k)} - \mathbf{u}^{(k)} \lambda_{,\alpha\beta}^{(k)} &= -\mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} - (\mathbf{A}_{,\alpha} - \lambda_{,\alpha}^{(k)} \mathbf{I})_{,\beta} \mathbf{u}_{,\beta}^{(k)} \\ &\quad - (\mathbf{A}_{,\beta} - \lambda_{,\beta}^{(k)} \mathbf{I})_{,\alpha} \mathbf{u}_{,\alpha}^{(k)} \end{aligned} \quad (2.3.29)$$

or, in partitioned matrix form,

$$\begin{aligned}
 [\mathbf{A} - \lambda^{(k)} \mathbf{I}] \begin{Bmatrix} -\mathbf{u}^{(k)} \\ \frac{\mathbf{u}_{,\alpha\beta}^{(k)}}{\lambda_{,\alpha\beta}^{(k)}} \end{Bmatrix} &= -\mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} - (\mathbf{A}_{,\alpha} - \lambda_{,\alpha}^{(k)} \mathbf{I}) \mathbf{u}_{,\beta}^{(k)} \\
 &\quad - (\mathbf{A}_{,\beta} - \lambda_{,\beta}^{(k)} \mathbf{I}) \mathbf{u}_{,\alpha}^{(k)}
 \end{aligned} \tag{2.3.30}$$

Following the same reasoning as before, eq. (2.3.30) is written as

$$\mathbf{B} \mathbf{y}_2 = \mathbf{s} \tag{2.3.31}$$

where

$$\mathbf{y}_2 = \begin{Bmatrix} \mathbf{u}_{,\alpha\beta}^{(k)} \\ -\lambda_{,\alpha\beta}^{(k)} \end{Bmatrix} \text{ with } m\text{-th element deleted}$$

$$\mathbf{s} = -\mathbf{A}_{,\alpha\beta} \mathbf{u}^{(k)} - (\mathbf{A}_{,\alpha} - \lambda_{,\alpha}^{(k)} \mathbf{I}) \mathbf{u}_{,\beta}^{(k)} - (\mathbf{A}_{,\beta} - \lambda_{,\beta}^{(k)} \mathbf{I}) \mathbf{u}_{,\alpha}^{(k)} \tag{2.3.32}$$

Note that, if $\lambda^{(k)}$ is a distinct eigenvalue of \mathbf{A} and if $u_m^{(k)} \neq 0$, then the matrix \mathbf{A} is of rank $(n - 1)$ and the m -th column that is deleted is linearly dependent on the other columns. Hence the matrix \mathbf{B} is non-singular. The matrix \mathbf{B} will also be well-conditioned if $u_m^{(k)}$ is the largest component in the eigenvector $\mathbf{u}^{(k)}$ and the matrix \mathbf{A} is itself not ill-conditioned. The vectors \mathbf{y}_1 and \mathbf{y}_2 can be obtained by standard solution methods. If the matrix \mathbf{A} is banded or if the derivatives of both right and left eigenvectors are required, it may be more efficient to use a partitioning scheme as described below.

2.3.3 Modification of Direct Method for Banded Matrices

Equations (2.3.27) and (2.3.31) can be written as

$$(\mathbf{A} - \lambda^{(k)}\mathbf{I})_{m\text{-th column deleted}, \alpha} \mathbf{u}_{, \alpha}^{(k)} - \lambda_{, \alpha}^{(k)} \mathbf{u}^{(k)} = \mathbf{r} \quad (2.3.33)$$

Let $\mathbf{u}^{(k)}$ be normalized so that $u_m^{(k)} = u_{m0}^{(k)} = \text{constant}$

Eq. (2.3.33) is a system of n equations. Writing the m -th equation separately, we have, if the superscript (k) is omitted for notational convenience,

$$\mathbf{C} \mathbf{x}_{, \alpha} - \lambda_{, \alpha} \mathbf{x} = \mathbf{t} \quad (2.3.34)$$

and

$$\mathbf{a}_m^T \mathbf{x}_{, \alpha} - \lambda_{, \alpha} u_{m0} = r_m \quad (2.3.35)$$

where

$$\mathbf{C} = (\mathbf{A} - \lambda \mathbf{I})_{m\text{-th row and column deleted}}$$

$$\mathbf{x}_{, \alpha} = \mathbf{u}_{, \alpha} \text{ } m\text{-th row deleted}$$

$$\mathbf{x} = \mathbf{u} \text{ } m\text{-th row deleted}$$

$$\mathbf{t} = \mathbf{r} \text{ } m\text{-th row deleted}$$

$\mathbf{a}_m^T = m$ -th row of \mathbf{A} with the m -th column deleted

From (2.3.35),

$$\lambda_{,\alpha} u_{m0} = \mathbf{a}_m^T \mathbf{x}_{,\alpha} - r_m \quad (2.3.36)$$

From (2.3.34),

$$\mathbf{x}_{,\alpha} = \mathbf{C}^{-1}(\lambda_{,\alpha} \mathbf{x} + \mathbf{t}) \quad (2.3.37)$$

Eliminating $\mathbf{x}_{,\alpha}$, we have

$$\lambda_{,\alpha} = \frac{\mathbf{t}^T \mathbf{b}_m - r_m}{u_{m0} - \mathbf{x}^T \mathbf{b}_m} \quad (2.3.38)$$

where

$$\mathbf{b}_m = [\mathbf{C}^T]^{-1} \mathbf{a}_m$$

Proceeding in a similar manner for the left eigenvector,

$$\mathbf{y}_{,\alpha} = [\mathbf{C}^T]^{-1}(\lambda_{,\alpha} \mathbf{y} + \mathbf{t}) \quad (2.3.39)$$

where

$$\mathbf{y}_{,\alpha} = \mathbf{v}_{,\alpha} \text{ } m\text{-th row deleted}$$

$$\mathbf{y} = \mathbf{v} \text{ } m\text{-th row deleted}$$

$\mathbf{t}_j = (\mathbf{r}_j)_{m\text{-th row deleted}}$

\mathbf{r}_j being the appropriate right hand side.

Thus the following procedure can be used to obtain the derivatives $\lambda_{,\alpha}$ and $\mathbf{u}_{,\alpha}$.

1. Form a LU decomposition of the matrix \mathbf{C} .
2. Solve $\mathbf{b}_m = [\mathbf{C}^T]^{-1}\mathbf{a}_m$ by forward substitution.
3. Calculate $\lambda_{,\alpha}$ from (2.3.38).
4. Calculate $\mathbf{x}_{,\alpha}$ from (2.3.37) by backward substitution.
5. Expand $\mathbf{x}_{,\alpha}$ to $\mathbf{u}_{,\alpha}$ setting $u_{m,\alpha} = 0$.

If the derivatives $\mathbf{v}_{,\alpha}$ of the left eigenvectors are also required, only three further steps are needed.

6. Calculate $\mathbf{y}_{,\alpha}$ from (2.3.39) by forward substitution.
7. Expand $\mathbf{y}_{,\alpha}$ to $\mathbf{v}_{,\alpha}$ setting $v_{m,\alpha} = 0$.
8. Normalize $\mathbf{v}_{,\alpha}$ appropriately depending on the normalization of \mathbf{v} . For example, to obtain the derivative of the left eigenvector that satisfies the normalization condition of eq. (2.2.3), subtract $(\mathbf{v}^T \mathbf{u}_{,\alpha} + \mathbf{v}_{,\alpha}^T \mathbf{u})\mathbf{v}$.

The matrix \mathbf{C} needs to be factored only once. Also, the matrix \mathbf{C} retains the bandedness characteristics of the original matrix \mathbf{A} , so that advantage can be taken of it. Furthermore, higher order derivatives can be obtained by merely substituting an appropriate right hand side vector, \mathbf{r} . However, higher order derivatives can suffer in accuracy because of accumulated round-off error.

The conditioning of matrix \mathbf{C} needs some comment. Note that \mathbf{C} is obtained from the singular matrix $(\mathbf{A} - \lambda^{(k)}\mathbf{I})$ by deleting both the row and column corresponding to index m . Hence, for matrix \mathbf{C} to be non-singular, one must make sure that the m -th row is linearly dependent on the other rows as well as that the m -th column is linearly dependent on the other columns. In other words, \mathbf{C} is non-singular iff $u_m^{(k)} \neq 0$ and $v_m^{(k)} \neq 0$. If $v_m^{(k)}$ is very small compared to the largest element in $\mathbf{v}^{(k)}$, steps 2 and 4 in the above procedure will give inaccurate results even if $u_m^{(k)}$ is the largest element in $\mathbf{u}^{(k)}$. In general, it is not possible to make a good choice for m without the knowledge of the left eigenvector. Since the calculation of left eigenvector using forward substitution in an inverse iteration scheme is cheap (as explained later in Section 3.1), it is suggested that the left eigenvector be calculated and the index m be chosen as in eq.(2.2.6). This is the same criterion used by Nelson[31] and will assure as well-conditioned a matrix \mathbf{C} as possible.

In summary, we note that, in calculating derivatives by Direct Method,

- left eigenvectors are not used.

- a complete solution of the eigenvalue problem is not required, if the derivatives of only a few of the eigenvalues and eigenvectors are sought. This is in contrast to Adjoint Method which requires all the left and right eigenvectors to calculate the first derivative of any eigenvector.
- calculation of any derivative requires the solution of a system of linear equations.
- only one matrix factorization needs to be performed for all orders of derivatives of an eigenvalue and its corresponding right and left eigenvectors.

2.3.4 Iterative Methods

Andrew[27] proposed an iterative algorithm to calculate the first derivatives of eigenvalues and eigenvectors. This algorithm is a refined and generalized version of the iterative scheme developed by Rudisill and Chu[24]. Except for the dominant eigenvalue, the convergence of this algorithm seems to be very much dependent on the choice of the initial values for the derivatives. To be efficient for non-hermitian matrices, this iterative method requires a complex eigenvalue shifting strategy which is not easy to implement. Hence this method is not considered.

Chapter 3

Efficiency Considerations in Calculating Derivatives

In order to establish criteria for the selection of the most efficient algorithm for calculating the derivatives of eigenvalues and eigenvectors in a given application, we compare the operation counts and actual CPU times required by Adjoint and Direct methods. The decision as to which algorithm is best is necessarily problem-dependent. The comparison is, however, described in terms of three variables that can usually be ascribed to a given problem and which significantly influence the decision. These variables are

1. the size of the matrix n
2. the number of design parameters m
3. the number of eigenvalues of interest l .

3.1 Operation Counts

To start with, let us consider the operation counts (multiplications and divisions only) for the adjoint methods given by eqs. (2.3.1),(2.3.8)-(2.3.18) and the direct methods given by eqs.(2.3.27)-(2.3.32). They are summarized in Table 1. It should be noted that the operation counts represent an estimate of the actual number of operations performed by a solution routine and include only the most significant terms. The actual number of operations will vary slightly depending on programming details. The effect of the sparsity of the matrix derivative $A_{,\alpha}$ is modeled in the operation counts by the parameter κ , defined such that the number of operations in evaluating the product $A_{,\alpha}u$ is equal to κn^2 (that is, $\kappa = 1$ corresponds to a full $A_{,\alpha}$).

The eigenvalues are calculated using the EISPACK subroutine package[49] by first reducing the matrix to upper hessenberg form using unitary similarity transformations and then applying the QR algorithm. The number of operations and the CPU time for calculating the eigenvalues is not relevant in evaluating the methods to calculate the derivatives. The operation count for eigenvalue computation is given only for comparison.

The right eigenvectors are calculated by inverse iteration on the same upper hessenberg matrix used for calculating the eigenvalues and are back transformed using standard subroutines in the package EISPACK. The corresponding operation count is given in Table 1. The inverse iteration algorithm is an extremely powerful method for computing eigenvectors and is

much superior in accuracy as well as speed of convergence to the common alternative algorithms based on the solution of homogeneous equations or direct iteration. Algorithms based on the solution of homogeneous equations are limited in their accuracy by the accuracy of the eigenvalue and those based on direct iteration are limited in their convergence, particularly for eigenvectors not corresponding to either the largest or the smallest eigenvalue.

For the calculation of left eigenvectors, it is important to note that there is no need to repeat the process with the transposed matrix. The left eigenvectors are obtained cheaply using forward substitution in place of backward substitution in the inverse iteration process. There is also no need to repeat the matrix factorization. A subroutine was written to calculate the right and left eigenvectors in this manner and the corresponding operation count is given in Table 1.

Table 1 gives the operation count of evaluating the individual steps. To obtain the number of operations involved in evaluating the derivatives, we must add the operation counts for all the steps required in the calculations. These counts are given in the following discussion.

Table 1. Operation Counts

Eigenvalues and Eigenvectors	
Operation	Operation Count
Evaluation of eigenvalues	$8n^3$ to $10n^3$
Evaluation of right eigenvectors	$l(2n^2)$
Evaluation of left eigenvectors	$l(\frac{3}{2}n^2)$
Adjoint Methods	
Operation	Operation Count
Evaluation of eq. (2.3.1)	$lmn^2\kappa$
Evaluation of eq. (2.3.8), (2.3.10),(2.3.11)	$lmn^2(\kappa + 2)$
Evaluation of eq. (2.3.18)	$l \binom{m}{2} n^2\kappa$
Direct Methods	
Operation	Operation Count
LU decomposition of matrix B	$l(\frac{n^3}{3})$
Formulation and solution of eq.(2.3.27)	$lmn^2(\kappa + 1)$
Formulation and solution of eq.(2.3.31)	$l \binom{m}{2} n^2(3\kappa + 1)$

3.2 CPU Time Statistics

In the following tables, computational cost for the calculation of the first and second derivatives of eigensystems are compared for matrices of order 20, 40 and 60. The CPU time statistics are obtained on the IBM 3084 computer using the VS-FORTRAN compiler with no compiler optimization. The correlation between operation counts and CPU times is shown in Tables 2 and 3. The ratio of operation count(OC) and CPU time for various operations, tabulated in Tables 2 and 3, is about 10^5 operations per CPU second with a variability of 27 percent.

The typical matrices are generated for the dynamic stability aeroelastic analysis of a compressor stage rotor with mistuned blades. The geometric and structural parameters of the rotor and formulation and method of analysis are the same as those of NASA Test Rotor 12 described in reference[50] except that the number of blades and the torsional frequencies are varied. The torsional frequency values are selected randomly from a population of mean 1.0 and standard deviation 0.01. The standard deviations of the actual samples are slightly different.

Table 2. Correlation between Operation Count(OC) and CPU time

Calculation of right eigenvectors			
<i>n</i>	<i>l</i>		OC/CPU seconds ($\times 10^4$)
60	60		8.6
60	10		8.3
Calculation of right and left eigenvectors			
<i>n</i>	<i>l</i>		OC/CPU seconds ($\times 10^4$)
60	60		8.5
60	10		9.2
Evaluation of eq. (2.3.1)			
<i>n</i>	<i>l</i>	<i>m</i>	OC/CPU seconds ($\times 10^4$)
60	60	10	10.7
60	60	5	10.7
60	10	10	10.7
60	10	5	10.7
Evaluation of eq. (2.3.8),(2.3.10),(2.3.11)			
<i>n</i>	<i>l</i>	<i>m</i>	OC/CPU seconds ($\times 10^4$)
60	60	10	12.0
60	60	5	11.9
60	10	10	11.9
60	10	5	11.9

Table 3. Correlation between Operation Count(OC) and CPU time(Contd.)

Evaluation of eq. (2.3.18)			
<i>n</i>	<i>l</i>	<i>m</i>	OC/CPU seconds ($\times 10^4$)
60	60	10	9.1
60	10	5	7.4
Decomposition of matrix B			
<i>n</i>	<i>l</i>		OC/CPU seconds ($\times 10^4$)
60	60		8.5
60	10		9.2
Evaluation of eq. (2.3.27)			
<i>n</i>	<i>l</i>	<i>m</i>	OC/CPU seconds ($\times 10^4$)
60	60	10	12.9
60	10	5	13.0

3.3 Calculation of First derivatives of Eigenvalues only

Operation Count	
Adjoint Method	$l(\frac{7}{2}n^2 + \kappa mn^2)$
Direct Method	$l[\frac{n^3}{3} + (\kappa + 1)mn^2]$

It is clear from the operation count that the Adjoint Method, which is an $O(n^2)$ process, is superior to Direct Method, an $O(n^3)$ process, for large n . The number of design variables and the number of eigenvalues of interest have no bearing on this conclusion. As the order of the matrix increases, the direct method becomes more expensive. For example, for 5 design variables and 10 eigenvalues of interest, the CPU time for the Direct method is 2.3 times more expensive than for the Adjoint Method for $n = 20$, and for $n = 60$, the ratio is 3.0.

3.4 Calculation of First derivatives of Eigenvalues and Eigenvectors

Operation Count	
Adjoint Method	$\frac{7}{2}n^3 + lmn^2(\kappa + 2)$
Direct Method	$\frac{ln^3}{3} + lmn^2(\kappa + 1)$

When the derivatives of both eigenvalues and right eigenvectors are required, the choice of method is dependent on the values of l and m . When very few eigenvalues are of interest, the Direct method is cheaper. When many eigenvalues are of interest, the Direct method is more expensive than the Adjoint method. However, this effect of the number of eigenvalues of interest is less significant when the number of design variables is large. As the number of design variables increases, the direct method becomes more competitive, even when all eigenvalues are of interest. For a 60 x 60 full ($\kappa = 1$) matrix, this is illustrated in Figure 1 on page 40.

The operation count shows that the computation by adjoint method of eigenvector derivative, which is necessary for the second derivative of eigenvalue, is an $O(n^3)$ process and is more expensive than the computation

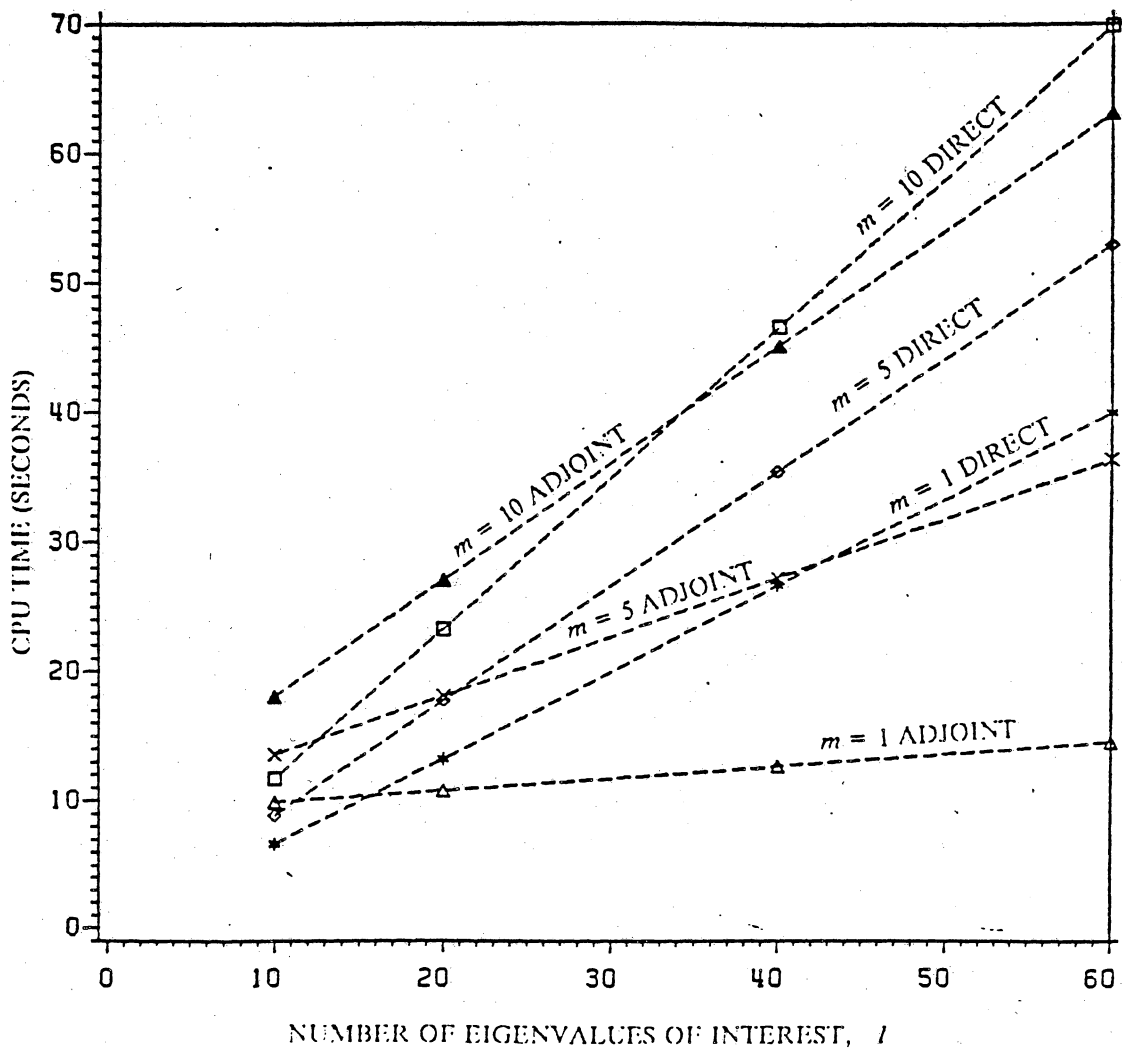


Figure 1. CPU Times for calculation of first derivatives of eigenvalues and eigenvectors for a 60 x 60 matrix

of the eigenvector itself which is an $O(n^2)$ process using the procedure described in Section 3.1. This fact is significant as some authors have stated the opposite[6,7].

3.5 Calculation of First and Second derivatives of Eigenvalues only

Operation Count	
Adjoint Method	$\frac{7}{2}n^3 + (\kappa + 1)mn^3 + l \binom{m}{2} n^{2\kappa}$
Direct Method	$l\frac{n^3}{3} + l \binom{m}{2} n^2(3\kappa + 1)$
Direct-Adjoint Method	$l\frac{n^3}{3} + l \binom{m}{2} n^{2\kappa} + lmn^2(2\kappa + 1)$

The Direct-Adjoint Method denotes the calculation of the eigenvector derivatives by the Direct method and the eigenvalue second derivatives by the Adjoint Method. The third term in the operation count for the Direct-Adjoint Method is significant only when m is small. From the operation count, it is seen that the Direct-Adjoint Method is always cheaper than the Direct Method. Hence, the choice lies between the Adjoint Method and the Direct-Adjoint method. Here, considerations similar to those of the last section hold and the choice of method depends on the values of l and m . When few eigenvalues are of interest, the Direct-Adjoint method is cheaper. When many eigenvalues are of interest, the Adjoint method is superior. But this advantage of Adjoint

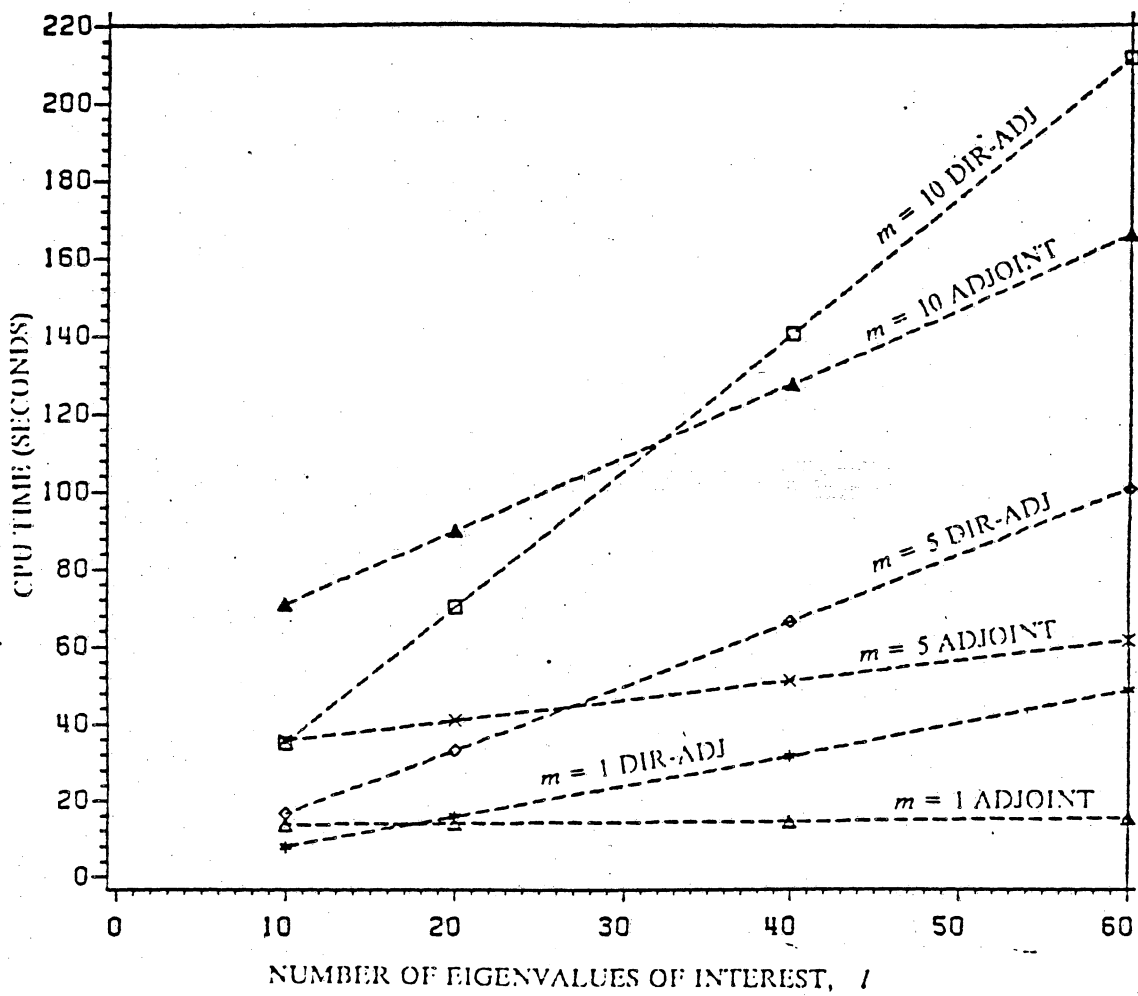


Figure 2. CPU Times for calculation of second derivatives of eigenvalues for a 60 x 60 matrix

Method diminishes as the number of design variables increases. This is again illustrated for a 60 x 60 full matrix ($\kappa = 1$) in Figure 2 on page 43.

Chapter 4

Approximate Eigenvalues of Modified Systems

4.1 Introduction

The eigenvalue problem to be solved is

$$\mathbf{A}\mathbf{u}^{(k)} = \lambda^{(k)}\mathbf{u}^{(k)} \quad (4.1.1)$$

and

$$\mathbf{v}^{(k)T}\mathbf{A} = \lambda^{(k)}\mathbf{v}^{(k)T} \quad (4.1.2)$$

where \mathbf{A} is a general complex matrix of order n and $\lambda^{(k)}$, $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ are the k -th eigenvalue and right and left eigenvectors respectively.

The matrix \mathbf{A} differs by a small amount from a nominal matrix \mathbf{A}_0 . The eigenvalues λ_0 and the right and left eigenvectors \mathbf{u}_0 and \mathbf{v}_0 are taken to be known.

$$\mathbf{A} = \mathbf{A}_0 + \Delta\mathbf{A} \quad (4.1.3)$$

$$\mathbf{A}_0\mathbf{u}_0^{(k)} = \lambda_0^{(k)}\mathbf{u}_0^{(k)} \quad (4.1.4)$$

and

$$\mathbf{v}_0^{(k)T}\mathbf{A}_0 = \lambda_0^{(k)}\mathbf{v}_0^{(k)T} \quad (4.1.5)$$

Throughout this chapter, we will also assume that the left eigenvectors are normalized such that

$$\mathbf{v}_0^{(k)T}\mathbf{u}_0^{(k)} = 1 \quad (4.1.6)$$

The eigenvalues λ and the eigenvectors \mathbf{u} and \mathbf{v} can be written as

$$\begin{aligned} \lambda^{(k)} &= \lambda_0^{(k)} + \Delta\lambda^{(k)} \\ \mathbf{u}^{(k)} &= \mathbf{u}_0^{(k)} + \Delta\mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} &= \mathbf{v}_0^{(k)} + \Delta\mathbf{v}^{(k)} \end{aligned} \quad (4.1.7)$$

Let $\mathbf{p}_{\alpha 0}$ be the vector of nominal design variables and $\Delta\mathbf{p}_{\alpha}$ be a vector of perturbations from the nominal design variables so that

$$\mathbf{A}_0 = \mathbf{A}(\mathbf{p}_{\alpha 0}) \quad (4.1.8)$$

and

$$\mathbf{A} = \mathbf{A}(\mathbf{p}_{\alpha 0} + \Delta \mathbf{p}_{\alpha}) \quad (4.1.9)$$

Approximate quantities are denoted by the subscript a. For example, an approximation for the eigenvalue $\lambda^{(k)}$ is denoted by $\lambda_a^{(k)}$. All derivatives are evaluated at the nominal design.

An exact relation exists between $\Delta \lambda$, $\Delta \mathbf{A}$, $\Delta \mathbf{u}$ and $\Delta \mathbf{v}$ as follows:

$$\lambda^{(k)} = \frac{\lambda_0^{(k)} + \mathbf{v}_0^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)} + \lambda_0^{(k)} (\mathbf{v}_0^{(k)T} \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \mathbf{u}_0) + \mathbf{v}_0^{(k)T} \Delta \mathbf{A} \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \mathbf{A}_0 \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)} + \Delta \mathbf{v}^{(k)T} \Delta \mathbf{A} \Delta \mathbf{u}^{(k)}}{1 + \Delta \mathbf{v}^{(k)T} \mathbf{u}_0 + \mathbf{v}_0^T \Delta \mathbf{u} + \Delta \mathbf{v}^{(k)T} \Delta \mathbf{u}^{(k)}} \quad (4.1.10)$$

The object is to obtain $\lambda^{(k)}$ without solving a full eigenvalue problem. $\lambda^{(k)}$ can be obtained exactly using eq. (4.1.10), if $\Delta \mathbf{u}$ and $\Delta \mathbf{v}$ are known. Since the exact values $\Delta \mathbf{u}$ and $\Delta \mathbf{v}$ cannot be obtained without solving a full eigenvalue problem, various approximations can be formed based on the above expression.

The approximations are broadly classified into

1. Derivative based approximations
2. Rayleigh-Quotient based approximations

3. Trace-theorem based approximations

4. Others

4.2 Derivative Based Approximations

Derivative based approximations are of special importance in optimization problems because first derivatives are required anyway in most optimization algorithms.

The most common of the derivative based approximations are based directly on truncated Taylor series. We will consider Linear and Quadratic approximations in this category. The enormous cost associated with the computation of any higher derivatives with multiple design variables effectively precludes the possibility of using higher order approximations based on the Taylor series.

4.2.1 Linear Approximation(LIN)

This is the simplest approximation to be considered in this work. Linear Approximation is obtained by truncating the Taylor series expansion for the eigenvalue after two terms.

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \sum_{\alpha=1}^m \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \Delta p_\alpha \quad (4.2.1)$$

A linear approximation is usually inadequate in terms of accuracy because eigenvalues are often highly non-linear functions of design variables. The linear approximation will be referred to herein as LIN approximation.

4.2.2 Quadratic Approximation(QUAD)

The Quadratic approximation is obtained by truncating the Taylor series expansion for the eigenvalue after three terms.

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \sum_{\alpha=1}^m \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \Delta p_\alpha + \frac{1}{2} \sum_{\alpha=1}^m \sum_{\beta=1}^m \frac{\partial^2 \lambda^{(k)}}{\partial p_\alpha \partial p_\beta} \Delta p_\alpha \Delta p_\beta \quad (4.2.2)$$

The quadratic approximation can be quite expensive for large orders of the matrix or large number of design variables. Miura and Schmit[14] used a simplified form of the quadratic approximation and found that, considering global efficiency, the higher cost of the quadratic approximation can sometimes offset the higher accuracy in an optimization problem. The quadratic approximation will be referred to herein as QUAD approximation.

Because of these efficiency considerations, attempts were made to improve the accuracy of the linear approximations through the use of intermediate variables with respect to which the eigenvalues may be nearly

linear. However, Miura and Schmit[14] concluded that such intermediate variables cannot be found for a general structural problem.

The accuracy of linear approximations can also be improved in another fashion. This is by introducing non-linearities without, however, introducing the second derivatives, which are expensive to calculate.

4.2.3 Conservative Approximation

In optimization applications, it is often desired to have a conservative approximation. For eigenvalue problems, this usually means underestimating the eigenvalues. Starnes and Haftka[51] proposed a hybrid approximation, which is a combination of a linear approximation and a reciprocal approximation (linear in $1/p_\alpha$) such that it is the most conservative combination of the two. Since this approximation is only applicable to real quantities, it is applied to the real part and/or the imaginary part of the eigenvalue as required in particular applications.

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \sum_{\alpha=1}^m B_\alpha \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \Delta p_\alpha \quad (4.2.3)$$

where

$$B_{\alpha} = \begin{cases} 1 & \text{if } \frac{\partial \lambda^{(k)}}{\partial p_{\alpha}} \leq 0 \\ \frac{p_{\alpha 0}}{p_{\alpha}} & \text{if } \frac{\partial \lambda^{(k)}}{\partial p_{\alpha}} > 0 \end{cases}$$

Even though this approximation is not, in general, more accurate than the linear approximation, it is popular because it is more conservative than the linear approximation and it is also convex.

4.2.4 Generalized Inverse Power Approximation

Non-linearities can also be introduced into the linear approximation in a more direct manner, as in the Generalized Inverse Power approximation, described by Prasad[52,53].

The linear approximation is first reformulated as

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \sum_{\alpha=1}^m \left[\frac{\partial \lambda^{(k)}}{\partial p_{\alpha}} / \frac{\partial \varphi_{\alpha}}{\partial p_{\alpha}} \right] \delta \varphi_{\alpha} \quad (4.2.4)$$

and the function φ_{α} is chosen as

$$\varphi_{\alpha} = \frac{p_{\alpha}^r}{r} \quad (4.2.5)$$

where r is any real number.

Prasad[52] also gave an alternate formulation for $r = 0$. The real number r is a controlling parameter for the approximation. There is no obvious choice for r in a general problem.

4.2.5 Generalized Hybrid Approximation

Woo[54] combined the concepts of the conservative approximation of Starnes and Haftka[51] and the Generalized Inverse Power approximation and defined a Generalized Hybrid approximation as

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \sum_{\alpha=1}^m \frac{\partial \lambda^{(k)}}{\partial p_\alpha} (p_\alpha - p_{\alpha 0}) \left(\frac{p_\alpha}{p_{\alpha 0}} \right)^r \quad (4.2.6)$$

where

$$r = \begin{cases} g & \text{if } \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \geq 0 \\ g - h & \text{if } \frac{\partial \lambda^{(k)}}{\partial p_\alpha} < 0 \end{cases}$$

g being a real number and h being a positive integer such that $g \geq 0$ and $g - h \leq -1$. The choice of g and h is again not obvious, though the fact that larger values for g and h make the approximation more conservative may provide some guideline. Since this approximation too is only applicable to real quantities, it is applied to the real part and/or the imaginary part of the eigenvalue as required in particular applications.

4.2.6 Reduction Method(RDN)

A different approach to derivative based approximations is based on the Taylor series approximation to the eigenvectors. This consists of reducing the original eigenvalue problem to a series of smaller order eigenproblems and is inspired by Noor's concept of global approximation vectors[55]. The concept of global approximation vectors is extended here to general matrices.

Noor's formulation is simplified by assuming that the eigenvectors can be treated as linear functions of the design variables. For the sake of simplicity, consider only one design variable. Let

$$\mathbf{u}_a^{(k)} = \mathbf{u}_0^{(k)} + \gamma \mathbf{u}_{,\alpha}^{(k)} = [\mathbf{u}_0^{(k)} \quad \mathbf{u}_{,\alpha}^{(k)}] \begin{Bmatrix} 1 \\ \gamma \end{Bmatrix} \quad (4.2.7)$$

and substituting eq. (4.2.7) in eq. (4.1.1), we get,

$$\mathbf{A}[\mathbf{u}_0^{(k)} \quad \mathbf{u}_{,\alpha}^{(k)}] \begin{Bmatrix} 1 \\ \gamma \end{Bmatrix} = \lambda_a [\mathbf{u}_0^{(k)} \quad \mathbf{u}_{,\alpha}^{(k)}] \begin{Bmatrix} 1 \\ \gamma \end{Bmatrix} \quad (4.2.8)$$

If the above equation is premultiplied by $[\mathbf{v}_0^{(k)} \quad \mathbf{v}_{,\alpha}^{(k)}]^T$, then we have

$$\mathbf{P} \begin{Bmatrix} 1 \\ \gamma \end{Bmatrix} = \lambda_a \mathbf{Q} \begin{Bmatrix} 1 \\ \gamma \end{Bmatrix} \quad (4.2.9)$$

where

$$\mathbf{P} = [\mathbf{v}_0^{(k)} \quad \mathbf{v}_{,\alpha}^{(k)}]^T \mathbf{A} [\mathbf{u}_0^{(k)} \quad \mathbf{u}_{,\alpha}^{(k)}]$$

and

$$Q = [v_0^{(k)} \quad v_{,\alpha}^{(k)}]^T [u_0^{(k)} \quad u_{,\alpha}^{(k)}]$$

Eq. (4.2.9) is a 2 x 2 linear eigenvalue problem and can be solved almost effortlessly. Of the two eigenvalues of eq. (4.2.9), the one closest to the linear approximation is chosen. In the case of multiple design variables, the only change needed is to replace $v_{,\alpha}^{(k)}$ and $u_{,\alpha}^{(k)}$ by the respective derivatives in the direction of change in design. It can be proved that, if the eigenvectors are linear functions of the design variable, eq. (4.2.9) gives an exact eigenvalue when u_0 and v_0 are normalized such that their first derivatives are respectively orthogonal to them. This approximation will be referred to herein as the RDN approximation.

4.3 Rayleigh Quotient Based Approximations

The Rayleigh quotient for the general eigenproblem given by eqs. (4.1.1) and (4.1.2) is defined as

$$R(x, y) = \frac{y^T A x}{y^T x} \quad (4.3.1)$$

When all the eigenvalues of matrix A are distinct, then the Rayleigh quotient $R(x, y)$ has a stationary value at $x = u^{(k)}$ and $y = v^{(k)}$ (the right and

left eigenvectors associated with the eigenvalue $\lambda^{(k)}$ for $k = 1, 2, \dots, N$. Further, this stationary value is equal to $\lambda^{(k)}$ [56]. That is,

$$R(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}) = \lambda^{(k)} = \frac{\mathbf{v}^{(k)T} \mathbf{A} \mathbf{u}^{(k)}}{\mathbf{v}^{(k)T} \mathbf{u}^{(k)}} \quad (4.3.2)$$

The approximations in this section are based on the above property. They seek to approximate the eigenvectors $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ and use them in the Rayleigh quotient to calculate the approximate eigenvalue.

4.3.1 Rayleigh Quotient with Nominal Eigenvectors(RAL1)

This approximation is obtained by simply using the nominal eigenvectors in the Rayleigh quotient.

$$\lambda_a^{(k)} = \mathbf{v}_0^{(k)T} \mathbf{A} \mathbf{u}_0^{(k)} \quad (4.3.3)$$

Equivalently,

$$\lambda_a^{(k)} = \lambda_0^{(k)} + \mathbf{v}_0^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)} \quad (4.3.4)$$

The second expression is cheaper to compute when $\Delta \mathbf{A}$ is sparse. We will refer to this approximation as the RAL1 approximation.

4.3.2 Rayleigh Quotient with Linearly Approximated Eigenvectors(RAL2)

The left and right eigenvectors are approximated from those of the nominal matrix using a 2-term Taylor series:

$$\begin{aligned}\mathbf{u}_a^{(k)} &= \mathbf{u}_0^{(k)} + \sum_{\alpha=1}^m \mathbf{u}_{,\alpha}^{(k)} \Delta p_\alpha \\ \mathbf{v}_a^{(k)} &= \mathbf{v}_0^{(k)} + \sum_{\alpha=1}^m \mathbf{v}_{,\alpha}^{(k)} \Delta p_\alpha\end{aligned}\tag{4.3.5}$$

The computation of the eigenvector derivatives $\mathbf{u}_{,\alpha}^{(k)}$ and $\mathbf{v}_{,\alpha}^{(k)}$ is discussed in Chapters 2 and 3.

These linearly approximated eigenvectors are then used in a Rayleigh quotient to generate an approximate eigenvalue. Our RAL2 approximation is therefore given by

$$\lambda_a^{(k)} = \frac{\mathbf{v}_a^{(k)T} \mathbf{A} \mathbf{u}_a^{(k)}}{\mathbf{v}_a^{(k)T} \mathbf{u}_a^{(k)}}\tag{4.3.6}$$

4.3.3 Rayleigh Quotient with Perturbed Eigenvectors(RAL3)

In this algorithm, the perturbation $\Delta \mathbf{A}$ in the matrix is used to evaluate the perturbations in the right and the left eigenvectors either by Adjoint or Direct

Method, which are analogous to the Adjoint and Direct methods described in Chapter 2. In the Adjoint method, assuming that the eigenvectors are normalized according to eq. (2.2.3) and (2.2.4), the perturbations in the right and left eigenvectors, $\delta \mathbf{u}^{(k)}$ and $\delta \mathbf{v}^{(k)}$ respectively, are calculated as follows:

$$\delta \mathbf{u}^{(k)} = \sum_{j=1}^n e_{kj} \mathbf{u}_0^{(j)} \quad \text{and} \quad \delta \mathbf{v}^{(k)} = \sum_{j=1}^n f_{kj} \mathbf{v}_0^{(j)} \quad (4.3.8)$$

where

$$e_{kj} = \frac{\mathbf{v}_0^{(j)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)}}{(\lambda_0^{(k)} - \lambda_0^{(j)})} \quad k \neq j \quad (4.3.9)$$

$$f_{kj} = \frac{\mathbf{v}_0^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(j)}}{(\lambda_0^{(k)} - \lambda_0^{(j)})} \quad k \neq j \quad (4.3.10)$$

and

$$e_{kk} = -f_{kk} = - \sum_{\substack{j=1 \\ j \neq k}}^n e_{kj} u_{0m}^{(j)} \quad (4.3.11)$$

McCalley[57] used this approach for error analysis of real symmetric eigenvalue problems. Chen and Wada[58,59] and Chen and Garba[60] used an equivalent formulation for real symmetric matrices and obtained eigenvalue approximations for all eigenvalues of the matrix simultaneously. Faddeev and Faddeeva[61] applied this approach to general matrices to improve the

accuracy of approximate eigenvalues. Meirovitch and Ryland[62] presented an extension to second order perturbations for general matrices.

In the Direct method, assuming that the right eigenvectors are normalized according to eq. (2.2.4), the perturbations in eigenvectors are calculated as follows. In eq. (4.1.1), substitute eq. (4.1.3) and

$$\begin{aligned}\lambda^{(k)} &= \lambda_0^{(k)} + \delta\lambda^{(k)} \\ \mathbf{u}^{(k)} &= \mathbf{u}_0^{(k)} + \delta\mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} &= \mathbf{v}_0^{(k)} + \delta\mathbf{v}^{(k)}\end{aligned}\tag{4.3.11}$$

to get, after ignoring second order perturbation terms,

$$\mathbf{A}_0\delta\mathbf{u}^{(k)} + \Delta\mathbf{A}\mathbf{u}_0^{(k)} = \lambda_0^{(k)}\delta\mathbf{u}^{(k)} + \delta\lambda^{(k)}\mathbf{u}_0^{(k)}\tag{4.3.12}$$

Eq. (4.3.12) is identical to eq. (2.3.24) when the derivatives are replaced by perturbations. Hence, the same solution methods described in Section 2.3 can be used to solve eq. (4.3.12) for the perturbations in the eigenvector. Thus,

$$\delta u_m^{(k)} = 0\tag{4.3.13}$$

and

$$\mathbf{B}_0\delta\mathbf{y} = \delta\mathbf{r}\tag{4.3.14}$$

where

$$\mathbf{B}_0 = [\mathbf{A}_0 - \lambda_0^{(k)}\mathbf{I} \mid -\mathbf{u}_0^{(k)}]_{m\text{-th column deleted}}$$

$$\delta\mathbf{y} = \left\{ \begin{array}{c} \delta\mathbf{u}^{(k)} \\ - \\ \delta\lambda^{(k)} \end{array} \right\} \text{ with } m\text{-th element deleted}$$

$$\mathbf{r} = -\Delta\mathbf{A}\mathbf{u}_0^{(k)} \quad (4.3.15)$$

The perturbations in left eigenvectors are obtained similarly using forward substitution.

The perturbations in the right and left eigenvectors are used to approximate the eigenvectors. Thus,

$$\mathbf{u}_a^{(k)} = \mathbf{u}_0^{(k)} + \delta\mathbf{u}^{(k)}$$

$$\mathbf{v}_a^{(k)} = \mathbf{v}_0^{(k)} + \delta\mathbf{v}^{(k)} \quad (4.3.16)$$

The approximate eigenvectors are then used in the Rayleigh quotient as in eq. (4.3.6) to form an approximation to the eigenvalue. Romstad, et. al [63] presented both the Adjoint method approach and an equivalent Direct method approach of this approximation for real symmetric matrices. However, they obtained different numerical results with the two approaches because of an error in their expressions for eigenvector perturbation by Adjoint method.

4.3.4 Rayleigh Quotient with One-step Inverse Iteration(RAL4)

The Inverse Iteration method has been recognized as a powerful tool for accurate computation of eigenvectors[47,64]. The unusual feature of the Inverse Iteration method is that accurate eigenvectors can be computed even when the eigenvalue is not known accurately as long as the eigenvalue is close enough to the correct eigenvalue. This feature can be used effectively to improve the accuracy of a rapid but rough approximation. In addition, a one-step inverse iteration is usually sufficient because most of the improvement in accuracy normally occurs in the first step and in a modification problem, the nominal eigenvector is available and provides an excellent initial iterate.

In this algorithm, a first approximation $\lambda_a^{(k)}$ to the eigenvalue $\lambda^{(k)}$ is formed as a Rayleigh Quotient (RAL1) given by eq. (4.3.3). This approximation is then used in a one-step inverse iteration scheme to obtain approximate eigenvectors as follows:

$$\begin{aligned} \mathbf{u}_a^{(k)} &= (\mathbf{A} - \lambda_a^{(k)}\mathbf{I})^{-1} \mathbf{u}_0^{(k)} \\ \mathbf{v}_a^{(k)} &= (\mathbf{A}^T - \lambda_a^{(k)}\mathbf{I})^{-1} \mathbf{v}_0^{(k)} \end{aligned} \tag{4.3.17}$$

The approximate eigenvectors are then used in the Rayleigh quotient as in eq. (4.3.6) to form an approximation to the eigenvalue. We will refer to this approximation as RAL4.

Note that the evaluation of the $\mathbf{u}_a^{(k)}$ and $\mathbf{v}_a^{(k)}$ requires only one matrix factorization, since the second part of eq. (4.3.17) can be solved by forward substitution using the same factored matrix used in solving the first part.

4.4 Trace-Theorem Based Approximations

These approximations are based on well-known iterative methods for finding the roots of a polynomial. We apply these methods to the characteristic polynomial, $p(\lambda)$ of matrix \mathbf{A} using only one step of the iteration for the approximation.

The remarkable feature of the approximations in this section lies in the fact that the coefficients of the characteristic polynomial need not be calculated explicitly. This is achieved by employing the following result, known as the Trace Theorem[56].

Trace Theorem: If $p(\lambda) \neq 0$, then

$$f(\lambda) = \frac{p'(\lambda)}{p(\lambda)} = - \left[\text{Trace of } (\mathbf{A} - \lambda\mathbf{I})^{-1} \right] \quad (4.4.1)$$

where $p'(\lambda) = \frac{dp}{d\lambda}$.

4.4.1 One-step Newton-Raphson Iteration(NRT1)

Here, we employ the Newton-Raphson formula

$$\lambda_a^{(k)} = \lambda_{a1}^{(k)} - \frac{p(\lambda_{a1}^{(k)})}{p'(\lambda_{a1}^{(k)})} = \lambda_{a1}^{(k)} - \frac{1}{f(\lambda_{a1}^{(k)})} \quad (4.4.2)$$

where $\lambda_{a1}^{(k)}$ is an initial approximation for the eigenvalue $\lambda^{(k)}$.

Using the Trace Theorem, eq. (4.4.1), an approximation is formulated as

$$\lambda_a^{(k)} = \lambda_{a1}^{(k)} + \frac{1}{\text{Trace of } (\mathbf{A} - \lambda_{a1}^{(k)}\mathbf{I})^{-1}} \quad (4.4.3)$$

Note that this approximation requires a matrix inversion. The initial approximation $\lambda_{a1}^{(k)}$ is chosen as the Rayleigh quotient of eq. (4.3.3).

We will refer to this approximation as NRT1 approximation.

4.4.2 Refined One-step Newton-Raphson Iteration(NRT2)

In this approximation, we utilize the second derivative of $p(\lambda)$ to obtain a better approximation. To refine the Newton-Raphson Iteration using the second derivative, consider the truncated Taylor series expansion

$$0 = p(\lambda_a^{(k)}) = p(\lambda_{a1}^{(k)}) + p'(\lambda_{a1}^{(k)})(\lambda_a^{(k)} - \lambda_{a1}^{(k)}) + 0.5p''(\lambda_{a1}^{(k)})(\lambda_a^{(k)} - \lambda_{a1}^{(k)})^2 \quad (4.4.4)$$

Use eq. (4.4.2) to eliminate $\lambda_a^{(k)}$ in the third term on the right hand side, to get

$$0 = p(\lambda_{a1}^{(k)}) + p'(\lambda_{a1}^{(k)})(\lambda_a^{(k)} - \lambda_{a1}^{(k)}) + \frac{p''(\lambda_{a1}^{(k)})p^2(\lambda_{a1}^{(k)})}{2[p'(\lambda_{a1}^{(k)})]^2} \quad (4.4.5)$$

so that the refined Newton-Raphson formula is

$$\lambda_a^{(k)} = \lambda_{a1}^{(k)} - \frac{p(\lambda_{a1}^{(k)})}{p'(\lambda_{a1}^{(k)})} - \frac{p''(\lambda_{a1}^{(k)})p^2(\lambda_{a1}^{(k)})}{2[p'(\lambda_{a1}^{(k)})]^3} \quad (4.4.6)$$

From the definition of $f(\lambda)$, we have

$$p''(\lambda) = p(\lambda)[f'(\lambda) + f^2(\lambda)] \quad (4.4.7)$$

Now, differentiating eq. (4.4.1) with respect to λ , while noting that

$$\frac{d}{d\lambda} (\mathbf{A} - \lambda \mathbf{I})^{-1} = [(\mathbf{A} - \lambda \mathbf{I})^{-1}]^2 \quad (4.4.8)$$

we get,

$$f'(\lambda) = - \text{Trace of } [(\mathbf{A} - \lambda \mathbf{I})^{-1}]^2 \quad (4.4.9)$$

This expression is then used in the refined Newton-Raphson formula of eq. (4.4.6) to obtain our next approximation given below.

$$\lambda_a^{(k)} = \lambda_{a1}^{(k)} - \frac{f(\lambda_{a1}^{(k)}) + 3f^2(\lambda_{a1}^{(k)})}{2f^3(\lambda_{a1}^{(k)})} \quad (4.4.10)$$

Note that, in evaluating $f(\lambda)$ using eq. (4.4.9), the matrix multiplication need not be performed completely, since only the diagonal elements of the matrix product are needed.

This approximation will be referred to herein as the NRT2 approximation.

4.4.3 One-step Laguerre Iteration(LIT)

Laguerre iteration is often used to compute eigenvalues and is known to have excellent convergence properties[47]. For our purposes, we need only one step of the Laguerre iteration. The one-step Laguerre iteration consists of

$$\lambda_a = \lambda_{a1} - \frac{np(\lambda_{a1})}{p'(\lambda_{a1}) \pm [(n-1)^2 p'^2(\lambda_{a1}) - n(n-1)p(\lambda_{a1})p''(\lambda_{a1})]^{1/2}} \quad (4.4.11)$$

This approximation also utilizes the second derivative of $p(\lambda)$. The derivation of eq. (4.4.11) is given in Wilkinson[47]. To obtain our approximation, we rewrite the above, using the trace theorem, as

$$\lambda_a = \lambda_{a1} - \frac{n}{f(\lambda_{a1}) \pm [-(n-1)f^2(\lambda_{a1}) - n(n-1)f'(\lambda_{a1})]^{1/2}} \quad (4.4.12)$$

The evaluation of $f(\lambda)$ is described in Section 4.4.2. The sign in the denominator is chosen so as to make the denominator have the greater absolute value. We will refer to this approximation as the LIT approximation.

4.5 Other Approximations

An alternative approach to approximate eigenvalues is taken by Paipetis and Croustalis[65] who developed an algorithm to approximate the coefficients of the characteristic polynomial which is then solved to obtain approximate eigenvalues. In addition to the numerical difficulties associated with the evaluation of the coefficients of the characteristic polynomial and its solution, this method severely restricts the characteristics of the system matrix.

We will consider one more approximation called [1,1]Pade approximation.

4.5.1 [1,1] Pade Approximation(PAD1)

We derive the [1,1] Pade approximation by geometrical construction along the lines of Johnson[66]. Let us for the moment assume that the eigenvalue to be approximated is real.

Let $\lambda_{a1}^{(k)}$ and $\lambda_{a2}^{(k)}$ be the first and second approximations to the eigenvalue $\lambda^{(k)}$. The information contained in these approximations is exploited by Aitken's method[67] to obtain a hopefully better approximation. We form the differences

$$\delta\lambda_{a1}^{(k)} = \lambda_{a1}^{(k)} - \lambda_0^{(k)}$$

$$\delta\lambda_{a2}^{(k)} = \lambda_{a2}^{(k)} - \lambda_{a1}^{(k)} \tag{4.5.1}$$

If $\lambda_0^{(k)}$, $\lambda_{a1}^{(k)}$ and $\lambda_{a2}^{(k)}$ is a converging series, we will have

$$\delta\lambda_{a2} < \delta\lambda_{a1} \quad (4.5.2)$$

We now draw a straight line through the points $(\lambda_{a1}^{(k)}, \delta\lambda_{a1}^{(k)})$ and $(\lambda_{a2}^{(k)}, \delta\lambda_{a2}^{(k)})$ to extrapolate to $\delta\lambda^{(k)} = 0$ on a $(\lambda^{(k)}, \delta\lambda^{(k)})$ plot. This is illustrated in Figure 3. It is expected that the extrapolated value $\lambda_a^{(k)}$ would be a better approximation than either $\lambda_{a1}^{(k)}$ or $\lambda_{a2}^{(k)}$.

The result is

$$\lambda_a^{(k)} = \lambda_{a1}^{(k)} + \frac{\delta\lambda_{a1}^{(k)} \delta\lambda_{a2}^{(k)}}{\delta\lambda_{a1}^{(k)} - \delta\lambda_{a2}^{(k)}} \quad (4.5.3)$$

Although the motivation applied only to real eigenvalues, this result can be immediately extended to complex eigenvalues. Using eqs.(4.5.1), eq. (4.5.3) is rewritten as

$$\lambda_a^{(k)} = \frac{(\lambda_{a1}^{(k)})^2 - \lambda_0^{(k)}\lambda_{a2}^{(k)}}{2\lambda_{a1}^{(k)} - \lambda_0^{(k)} - \lambda_{a2}^{(k)}} \quad (4.5.4)$$

If the approximations $\lambda_{a1}^{(k)}$, $\lambda_{a2}^{(k)}$ are the linear and the quadratic approximations, the approximation of eq. (4.5.4) can be recognized to be in the form of the [1,1] Pade approximant. In the following, this is assumed and this approximation will be referred to as PAD1 approximation.

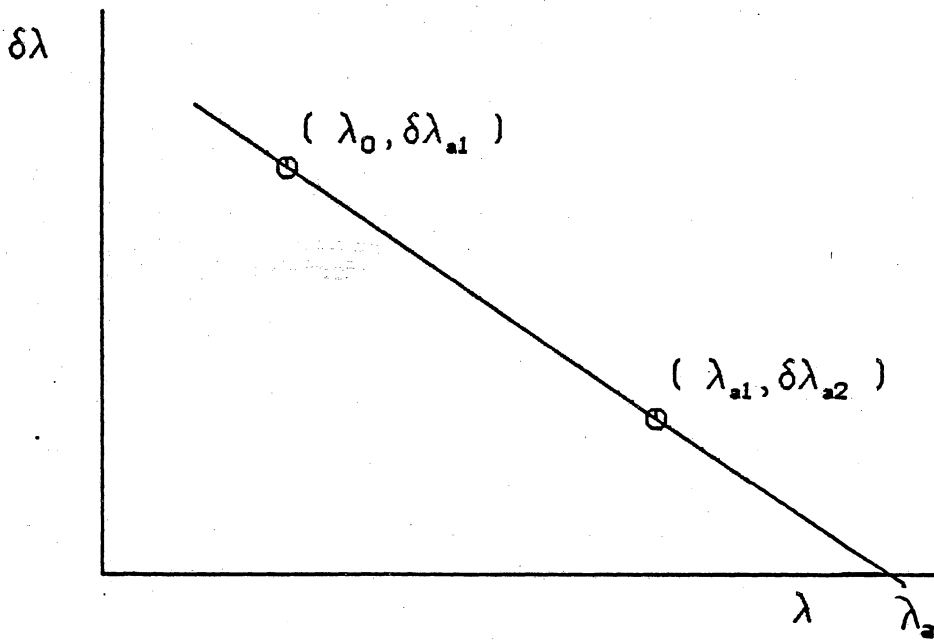


Figure 3. Derivation of [1,1] Padé Approximation using Geometric Construction

Chapter 5

Accuracy and Efficiency of Eigenvalue Approximations

5.1 Introduction

In Chapter 4, we listed several approximations for the eigenvalues of general matrices. For a given application, the selection of the appropriate approximation usually depends on the saving of computational time that a given approximation entails. In the task of selecting a good approximation, information about the accuracy and the efficiency of computation of the approximations is essential. Accuracy and efficiency are not independent elements in the selection of an approximation algorithm. Poor accuracy usually translates into low efficiency in the global process.

In this Chapter, the accuracy and efficiency considerations relating to the approximations listed in the last chapter are discussed. The accuracy considerations are treated in Section 5.2 and the efficiency considerations in Section 5.3. The Conservative, Generalized Inverse Power and the Generalized Hybrid approximations are not studied as their accuracy is problem dependent and a general assessment is not feasible.

Some of the approximations discussed have been applied to symmetric matrices by researchers in structural dynamics. However, there exists no systematic comparison of accuracy and efficiency. To the best of the author's knowledge, the trace theorem based algorithms have never been used for approximating eigenvalues of modified systems in the structural dynamics literature.

5.2 Accuracy Considerations in Approximating Eigenvalues

For simplicity of notation, we consider a single design variable. It will be obvious, however, that the results of the error analysis are applicable to the case of multiple design variables as well.

5.2.1 Order of an Approximation

For a useful comparison of the various approximations, we define the order of an approximation as follows:

Definition: If an approximation λ_a to an eigenvalue λ is such that the error

$$(\lambda_a - \lambda) = O[(\Delta p_\alpha)^{s+1}] \text{ or } (\lambda_a - \lambda) \cong C(\Delta p_\alpha)^{s+1} \text{ for } \Delta p_\alpha \rightarrow 0 \quad (5.2.1)$$

then that approximation is said to be of s -th order.

Note that, for a rigorous estimate of the error in an approximation, information about both the order of the approximation as well as the proportionality constant C in eq. (5.2.1) is needed. However, the order of the approximation is usually the most important property of the approximation and the proportionality constant is useful only when comparing approximations of the same order. In this work, attention is focused on the order of the approximation and the proportionality constant is discussed only in examples.

5.2.2 First Order Approximations

Among the approximations described in the last chapter, the linear approximation (LIN) and the Rayleigh quotient with nominal eigenvectors (RAL1) are first order approximations.

It is easy to show that the Linear approximation is of first order. Consider the two term Taylor series for $\lambda^{(k)}$ with remainder given by

$$\lambda^{(k)} = \lambda_0^{(k)} + \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \Delta p_\alpha + \frac{1}{2} \frac{\partial^2 \lambda^{(k)}}{\partial^2 p_\alpha} (\zeta_\alpha) (\Delta p_\alpha)^2 \quad (5.2.2)$$

where $p_\alpha \leq \zeta_\alpha \leq p_\alpha + \Delta p_\alpha$.

Comparing this to the linear approximation, eq. (4.2.1), we have the error

$$\lambda_a^{(k)} - \lambda_{LIN}^{(k)} = O(\Delta p_\alpha^2) \quad (5.2.3)$$

Hence the linear approximation is a first order approximation.

To find the error in the RAL1 approximation, subtract eq. (4.3.4) from the exact expression of eq. (4.1.10). Ignoring the third order terms, we have

$$\lambda^{(k)} - \lambda_{RAL1}^{(k)} = \frac{\mathbf{v}_0^{(k)T} \Delta \mathbf{A} \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \mathbf{A}_0 \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)} - \mathbf{v}_0^{(k)T} \Delta \mathbf{A} \mathbf{u}_0^{(k)} (\mathbf{v}_0^{(k)T} \Delta \mathbf{u}^{(k)} + \Delta \mathbf{v}^{(k)T} \mathbf{u}_0^{(k)}) - \lambda_0 \Delta \mathbf{v}^{(k)T} \Delta \mathbf{u}^{(k)}}{1 + \Delta \mathbf{v}^{(k)T} \mathbf{u}_0 + \mathbf{v}_0^T \Delta \mathbf{u} + \Delta \mathbf{v}^{(k)T} \Delta \mathbf{u}^{(k)}} \quad (5.2.4)$$

Considering that, to the first order,

$$\Delta \mathbf{A}, \Delta \mathbf{u}, \Delta \mathbf{v} = O(\Delta p_\alpha)$$

we have

$$\lambda^{(k)} - \lambda_{RAL1}^{(k)} = O(\Delta p_\alpha^2) \quad (5.2.5)$$

Thus, RAL1 is a first order approximation.

5.2.3 Second Order Approximations

The Quadratic approximation and its improvement by [1,1] Pade approximation(PAD1) are the second order approximations we described. To show that the quadratic approximation is of second order, consider the three term Taylor series with remainder given by

$$\lambda^{(k)} = \lambda_0^{(k)} + \frac{\partial \lambda^{(k)}}{\partial p_\alpha} \Delta p_\alpha + \frac{1}{2} \frac{\partial^2 \lambda^{(k)}}{\partial p_\alpha^2} (\Delta p_\alpha)^2 + \frac{1}{6} \frac{\partial^3 \lambda^{(k)}}{\partial p_\alpha^3} (\zeta_\alpha) (\Delta p_\alpha)^3 \quad (5.2.6)$$

Comparing this to eq. (4.2.2), we have the error in the quadratic approximation as

$$\lambda_a^{(k)} - \lambda_{QUAD}^{(k)} = O(\Delta p_\alpha^3) \quad (5.2.7)$$

showing that the quadratic approximation is a second order approximation. The PAD1 approximation is an improvement on the quadratic approximation and so it is at least of second order.

The linear, quadratic and the PAD1 approximations are applicable to all functions and do not take advantage of the special properties of the eigenvalue problem. All the other approximations we are going to discuss are developed

specifically for approximating the eigenvalues and it will be shown that they achieve higher accuracy with less computational effort.

5.2.4 Third Order Approximations

The reduction method RDN, the Rayleigh quotient based methods RAL2 and RAL3 and the one-step Newton-Raphson iteration NRT1 are the third order methods of approximation that we considered. As the first three approximations RDN, RAL2 and RAL3 are closely related, we will derive the order of only the RAL2 approximation and infer the orders of the RAL3 and the RDN from this derivation.

Recall that in the RAL2 algorithm, the eigenvalue is approximated by using linearly approximated left and right eigenvectors in the Rayleigh quotient. Hence, we may write the two term Taylor series with remainder for the eigenvectors as

$$\begin{aligned} \mathbf{u}^{(k)} &= \mathbf{u}_a^{(k)} + \frac{1}{2} \mathbf{u}_{,\alpha\alpha}^{(k)}(\zeta_\alpha)(\Delta p_\alpha)^2 \\ \mathbf{v}^{(k)} &= \mathbf{v}_a^{(k)} + \frac{1}{2} \mathbf{v}_{,\alpha\alpha}^{(k)}(\zeta_\alpha)(\Delta p_\alpha)^2 \end{aligned} \tag{5.2.8}$$

Hence, the approximate eigenvectors can be rewritten in terms of the exact eigenvectors as

$$\mathbf{u}_a^{(k)} = \mathbf{u}^{(k)} - \frac{1}{2} \mathbf{u}_{,\alpha\alpha}^{(k)}(\zeta_\alpha)(\Delta p_\alpha)^2$$

$$\mathbf{v}_a^{(k)} = \mathbf{v}^{(k)} - \frac{1}{2} \mathbf{v}_{,\alpha\alpha}^{(k)}(\zeta_\alpha)(\Delta p_\alpha)^2 \quad (5.2.9)$$

Substituting these expressions in eq. (4.3.6) and using the eqs. (4.1.1-2), we get

$$\lambda_a^{(k)} = \frac{\mathbf{v}_a^{(k)T} \mathbf{A} \mathbf{u}_a^{(k)}}{\mathbf{v}_a^{(k)T} \mathbf{u}_a^{(k)}} = \frac{\lambda^{(k)}(1 - Z^{(k)}) + O(\Delta p_\alpha^4)}{1 - Z^{(k)} + O(\Delta p_\alpha^4)} \quad (5.2.10)$$

where

$$Z^{(k)} = \frac{1}{2} \left[\mathbf{v}_{,\alpha\alpha}^{(k)T}(\zeta_\alpha) \mathbf{u}^{(k)} + \mathbf{v}^{(k)T} \mathbf{u}_{,\alpha\alpha}^{(k)}(\zeta_\alpha) \right] (\Delta p_\alpha)^2 \quad (5.2.11)$$

Carrying out the long division and putting $\lambda_a^{(k)} = \lambda_{RAL2}^{(k)}$, we find the error in the RAL2 approximation as

$$\lambda^{(k)} - \lambda_{RAL2}^{(k)} = O(\Delta p_\alpha^4) \quad (5.2.12)$$

It is hence proved that the RAL2 approximation is of third order.

It may be recalled that in the RAL3 approximation, we approximate the eigenvalue by a Rayleigh quotient using left and right eigenvectors that were obtained by using first order perturbations whereas in the RAL2 algorithm, we approximated the eigenvectors using their first derivatives in a two-term Taylor series. There is mathematically no difference between these two

methods and their difference lies only in the computational algorithms. It follows then that RAL2 is also a third order approximation. There is indeed little difference in the accuracy of the two approximations when they were tested on example matrices.

The order of the reduction method approximation(RDN) is difficult to obtain algebraically. However, we can infer the order of the RDN approximation by comparing it to the RAL2 approximation. We first note that the chief characteristic of the RAL2 approximation is that the eigenvector of the modified matrix is assumed to be in the subspace consisting only of the linearly approximated eigenvector. The RDN approximation is more flexible in that the eigenvector of the modified matrix is assumed to be in the subspace consisting of the original eigenvector and its derivative. Thus, the RDN approximation can be expected to be somewhat better than the RAL2 approximation. Hence, the RDN and the RAL2 approximations are expected to be of the same order but possess a different proportionality constant in the sense of eq. (5.2.1). This conclusion is validated by several numerical experiments.

To derive the order of the one-step Newton-Raphson iteration(NRT1), we first write the Taylor's series for the characteristic polynomial, $p(\lambda)$ and its derivative, $p'(\lambda)$, as

$$p(\lambda_{a1}^{(k)}) = p(\lambda^{(k)}) + (\lambda_{a1}^{(k)} - \lambda^{(k)})p' + 0.5(\lambda_{a1}^{(k)} - \lambda^{(k)})^2 p'' + \dots \quad (5.2.13)$$

and

$$p'(\lambda_{a1}^{(k)}) = p' + (\lambda_{a1}^{(k)} - \lambda^{(k)})p'' + 0.5(\lambda_{a1}^{(k)} - \lambda^{(k)})^2 p''' + \dots \quad (5.2.14)$$

where primes denote derivatives evaluated at $\lambda^{(k)}$. Then the Newton-Raphson Algorithm of eq. (4.4.2), modified by subtracting $\lambda^{(k)}$ from both sides and noting that $p(\lambda^{(k)}) = 0$,

$$\lambda_a^{(k)} - \lambda^{(k)} = \frac{(\lambda_{a1}^{(k)} - \lambda^{(k)})p' + 0.5(\lambda_{a1}^{(k)} - \lambda^{(k)})^2 p'' + \dots}{p' + (\lambda_{a1}^{(k)} - \lambda^{(k)})p'' + \dots} \quad (5.2.15)$$

may be written as

$$\lambda_a^{(k)} - \lambda^{(k)} = \frac{0.5(\lambda_{a1}^{(k)} - \lambda^{(k)})^2 p'' + (1/3)(\lambda_{a1}^{(k)} - \lambda^{(k)})^3 p''' + \dots}{p' + (\lambda_{a1}^{(k)} - \lambda^{(k)})p'' + \dots} \quad (5.2.16)$$

giving

$$\lambda_a^{(k)} - \lambda^{(k)} \cong (\lambda_{a1}^{(k)} - \lambda^{(k)})^2 \frac{p''}{2p'} \quad (5.2.17)$$

We have chosen the initial approximation $\lambda_{a1}^{(k)}$ to be the RAL1 approximation, which has already been shown to be a first order approximation. Hence,

$$(\lambda_{a1}^{(k)} - \lambda^{(k)}) = O(\Delta p_a^2) \quad (5.2.18)$$

so that

$$(\lambda_{NRT1}^{(k)} - \lambda^{(k)}) = O(\Delta p_a^4) \quad (5.2.19)$$

establishing that the NRT1 approximation is a third order approximation.

5.2.5 Higher Order Approximations

The remaining approximations, RAL4, NRT2 and LIT are all fifth order approximations. We proceed to obtain the order of these approximations.

To get the order of the RAL4 approximation, we follow Ostrowski's approach[68] using, however, the simplifying assumption that all the eigenvalues are well-separated. Let \mathbf{U} and \mathbf{V} denote the matrices whose columns are the right eigenvectors \mathbf{u} and the left eigenvectors \mathbf{v} respectively of the matrix \mathbf{A} . Let Λ denote the diagonal matrix of eigenvalues. From the biorthogonal property of the left and right eigenvectors normalized as given by eq. (2.2.3), we have

$$\mathbf{V}^T \mathbf{A} \mathbf{U} = \Lambda \quad \text{and} \quad \mathbf{V}^T \mathbf{U} = \mathbf{I} \quad (5.2.20)$$

Define

$$\eta_0^{(k)} = \mathbf{V}^T \mathbf{u}_0^{(k)}, \quad \xi_0^{(k)} = \mathbf{U}^T \mathbf{v}_0^{(k)}, \quad \eta_a^{(k)} = \mathbf{V}^T \mathbf{u}_a^{(k)}, \quad \xi_a^{(k)} = \mathbf{U}^T \mathbf{v}_a^{(k)} \quad (5.2.21)$$

From eqs. (5.2.20), we have

$$\mathbf{V}^T(\mathbf{A} - \lambda_{a1}^{(k)}\mathbf{I})\mathbf{U} = \mathbf{\Lambda} - \lambda_{a1}^{(k)}\mathbf{I} \quad (5.2.22)$$

so that

$$(\mathbf{A} - \lambda_{a1}^{(k)}\mathbf{I})^{-1} = \mathbf{U}(\mathbf{\Lambda} - \lambda_{a1}^{(k)}\mathbf{I})^{-1}\mathbf{V}^T \quad (5.2.23)$$

Using the above and the definitions of eq. (5.2.21) after premultiplying the first part of eq. (4.3.17) by \mathbf{V}^T , we obtain the relation

$$\boldsymbol{\eta}_a^{(k)} = (\mathbf{\Lambda} - \lambda_{a1}^{(k)}\mathbf{I})^{-1} \boldsymbol{\eta}_0^{(k)} \quad (5.2.24)$$

From the second part of eq. (4.3.17), we can obtain, in a similar manner,

$$\boldsymbol{\xi}_a^{(k)} = (\mathbf{\Lambda} - \lambda_{a1}^{(k)}\mathbf{I})^{-1} \boldsymbol{\xi}_0^{(k)} \quad (5.2.25)$$

Using eqs. (5.2.20-21), the RAL4 approximation can now be written as

$$\lambda_a^{(k)} = \frac{\boldsymbol{\xi}_a^{(k)T} \mathbf{\Lambda} \boldsymbol{\eta}_a^{(k)}}{\boldsymbol{\xi}_a^{(k)T} \boldsymbol{\eta}_a^{(k)}} \quad (5.2.26)$$

Substituting the expressions (5.2.24) and (5.2.25) in eq. (5.2.26), we have

$$\lambda_a^{(k)} = \frac{\sum_{i=1}^n \frac{\lambda^{(i)} \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}}{\sum_{i=1}^n \frac{\xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}} \quad (5.2.27)$$

Subtracting the exact eigenvalue $\lambda^{(k)}$ from both sides of this expression, we get after considerable algebra,

$$\frac{\lambda_a^{(k)} - \lambda^{(k)}}{(\lambda_{a1}^{(k)} - \lambda^{(k)})^2} = \frac{\sum_{\substack{i=1 \\ i \neq k}}^n \frac{(\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}}{\xi_{0k}^{(k)} \eta_{0k}^{(k)} + \sum_{\substack{i=1 \\ i \neq k}}^n \eta_{0i}^{(k)} \xi_{0i}^{(k)} \frac{(\lambda^{(k)} - \lambda_{a1}^{(k)})^2}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}} \quad (5.2.28)$$

Then, if the eigenvalues are well separated and if $\lambda_{a1}^{(k)}$ is a reasonably close approximation of the eigenvalue $\lambda^{(k)}$, the second term in the denominator will be negligible compared to the first. So, we may write

$$\frac{\lambda_a^{(k)} - \lambda^{(k)}}{(\lambda_{a1}^{(k)} - \lambda^{(k)})^2} \cong \frac{\sum_{\substack{i=1 \\ i \neq k}}^n \frac{(\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}}{\xi_{0k}^{(k)} \eta_{0k}^{(k)}} \quad (5.2.29)$$

However, $\lambda_{a1}^{(k)}$ is taken to be the RAL1 approximation. Hence we have,

$$\lambda_{a1}^{(k)} = \frac{\xi_0^{(k)T} \Lambda \eta_0^{(k)}}{\xi_0^{(k)T} \eta_0^{(k)}} \quad (5.2.30)$$

Subtracting $\lambda^{(k)}$ from both sides as before,

$$\lambda_{a1}^{(k)} - \lambda^{(k)} = \frac{\sum_{\substack{i=1 \\ i \neq k}}^n (\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{\xi_{0k}^{(k)} \eta_{0k}^{(k)} + \sum_{\substack{i=1 \\ i \neq k}}^n \eta_{0i}^{(k)} \xi_{0i}^{(k)}} \quad (5.2.31)$$

Or,

$$\xi_{0k}^{(k)} \eta_{0k}^{(k)} = \frac{\sum_{\substack{i=1 \\ i \neq k}}^n (\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda_{a1}^{(k)} - \lambda^{(k)})} - \sum_{\substack{i=1 \\ i \neq k}}^n \xi_{0i}^{(k)} \eta_{0i}^{(k)} \quad (5.2.32)$$

When the eigenvalues are well-separated and $\lambda_{a1}^{(k)}$ is close to the exact eigenvalue $\lambda^{(k)}$, the first term is dominant, so that

$$\xi_{0k}^{(k)} \eta_{0k}^{(k)} \cong \frac{\sum_{\substack{i=1 \\ i \neq k}}^n (\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda_{a1}^{(k)} - \lambda^{(k)})} \quad (5.2.33)$$

Substituting eq. (5.2.33) in eq. (5.2.29),

$$\frac{\lambda_a^{(k)} - \lambda^{(k)}}{(\lambda_{a1}^{(k)} - \lambda^{(k)})^2} \cong (\lambda_{a1}^{(k)} - \lambda^{(k)}) \frac{\sum_{\substack{i=1 \\ i \neq k}}^n \frac{(\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}}{(\lambda^{(i)} - \lambda_{a1}^{(k)})^2}}{\sum_{\substack{i=1 \\ i \neq k}}^n (\lambda^{(i)} - \lambda^{(k)}) \xi_{0i}^{(k)} \eta_{0i}^{(k)}} \quad (5.2.34)$$

Now, putting $\lambda_a^{(k)} = \lambda_{RAL4}^{(k)}$ and $\lambda_{a1}^{(k)} = \lambda_{RAL1}^{(k)}$, we have

$$\lambda_{RAL4}^{(k)} - \lambda^{(k)} \propto (\lambda_{RAL1}^{(k)} - \lambda^{(k)})^3 \quad (5.2.35)$$

Since RAL1 is a first order approximation as shown in Section 5.2.2,

$$(\lambda_{RAL1}^{(k)} - \lambda^{(k)}) = O(\Delta p_\alpha^6) \quad (5.2.36)$$

proving that the RAL4 approximation is of fifth order.

The derivation of the order of the NRT2 approximation is analogous to that of NRT1 approximation given in the last section if eq. (4.4.10) is used in place of eq. (4.4.2). After considerable algebra, we get

$$\lambda_{NRT2}^{(k)} - \lambda^{(k)} = O[(\lambda_{a1}^{(k)} - \lambda^{(k)})^3] \quad (5.2.37)$$

For the Laguerre Method, Wilkinson[47] gives

$$\lambda_{LIT}^{(k)} - \lambda^{(k)} = O[(\lambda_{a1}^{(k)} - \lambda^{(k)})^3] \quad (5.2.38)$$

Since the initial approximation used in both these algorithms is of first order, we have

$$(\lambda_{NRT2}^{(k)} - \lambda^{(k)}) = O(\Delta p_\alpha^6) \quad (5.2.39)$$

$$(\lambda_{LIT}^{(k)} - \lambda^{(k)}) = O(\Delta p_\alpha^6) \quad (5.2.40)$$

so that both the NRT2 and the LIT approximations are fifth order approximations as stated.

5.2.6 Validation of the Theoretical Results

Taking logarithms of both sides of eq. (5.2.1), we have

$$\log(\lambda_a - \lambda) = (s + 1) \log(\Delta p_\alpha) + \text{constant} \quad (5.2.41)$$

so that if the error in an s -th order approximation is plotted against the change in design variable on log-log scale, one must obtain a straight line with slope $(s + 1)$.

The orders of approximations obtained in the last section are verified by numerical experiments, summarized in the following figures. In order to minimize the effect of round-off errors as much as possible, a small matrix of order 5 is used for validation of theoretical results. The matrix elements are the quadratic polynomials of a design variable where the coefficients are generated using a random number generator. The errors in the approximate eigenvalues are in comparison to the exact eigenvalue that is obtained by the QR algorithm and improved by using the eigenvectors computed by inverse

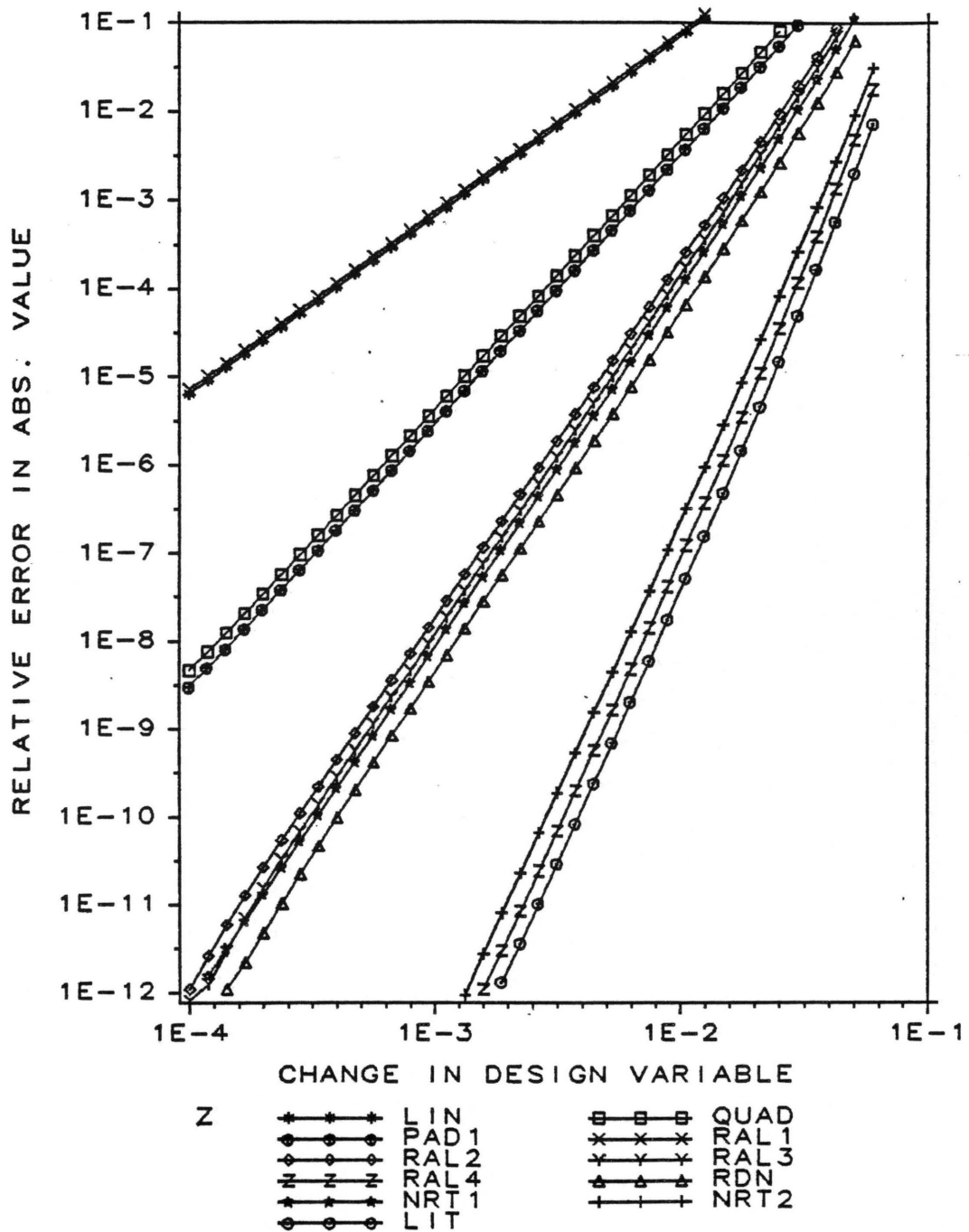


Figure 4. Relative Percentage errors in the Absolute value of the Eigenvalue 5 x 5 Matrix

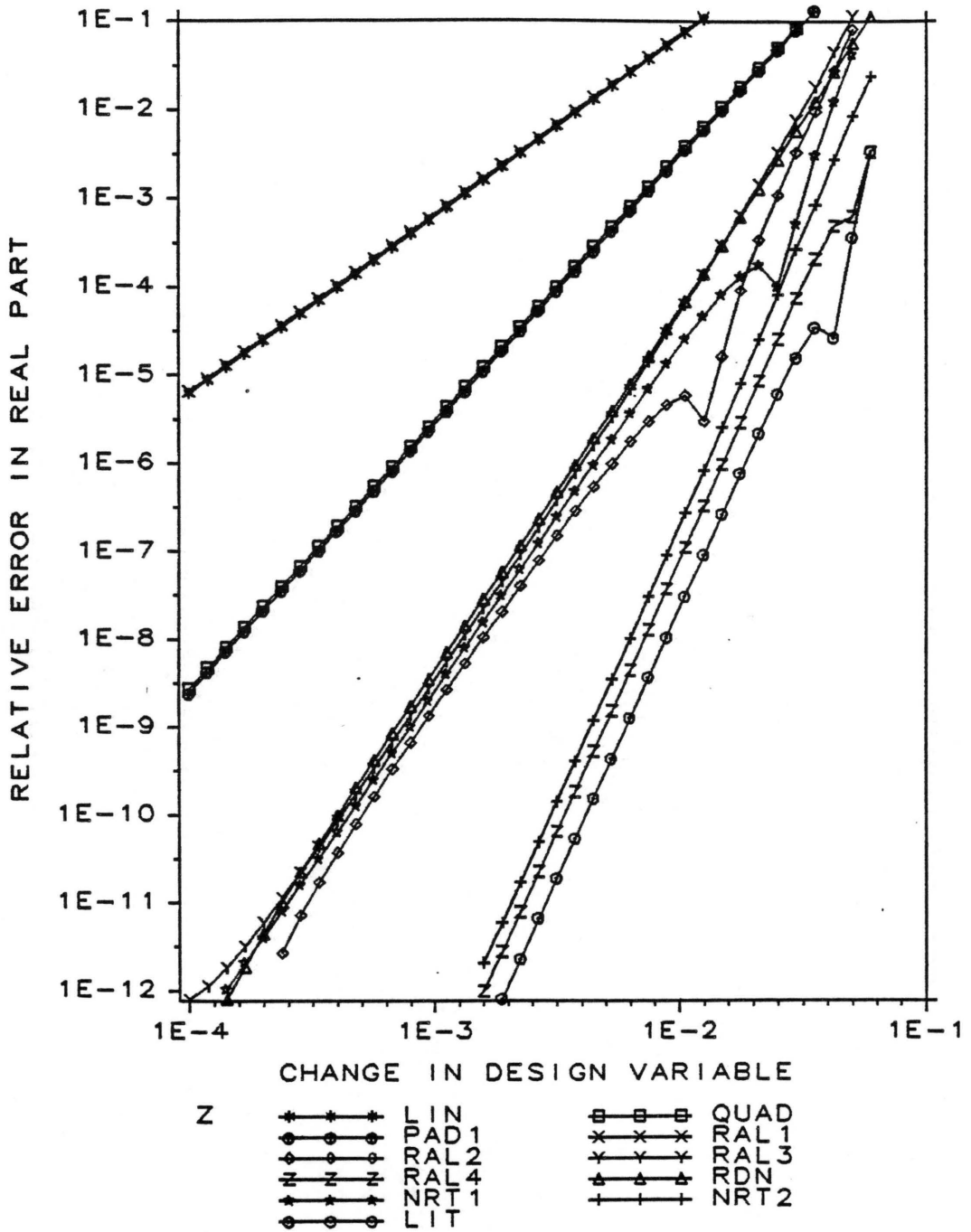


Figure 5. Relative Percentage errors in the Real Part of the Eigenvalue 5 x 5 Matrix

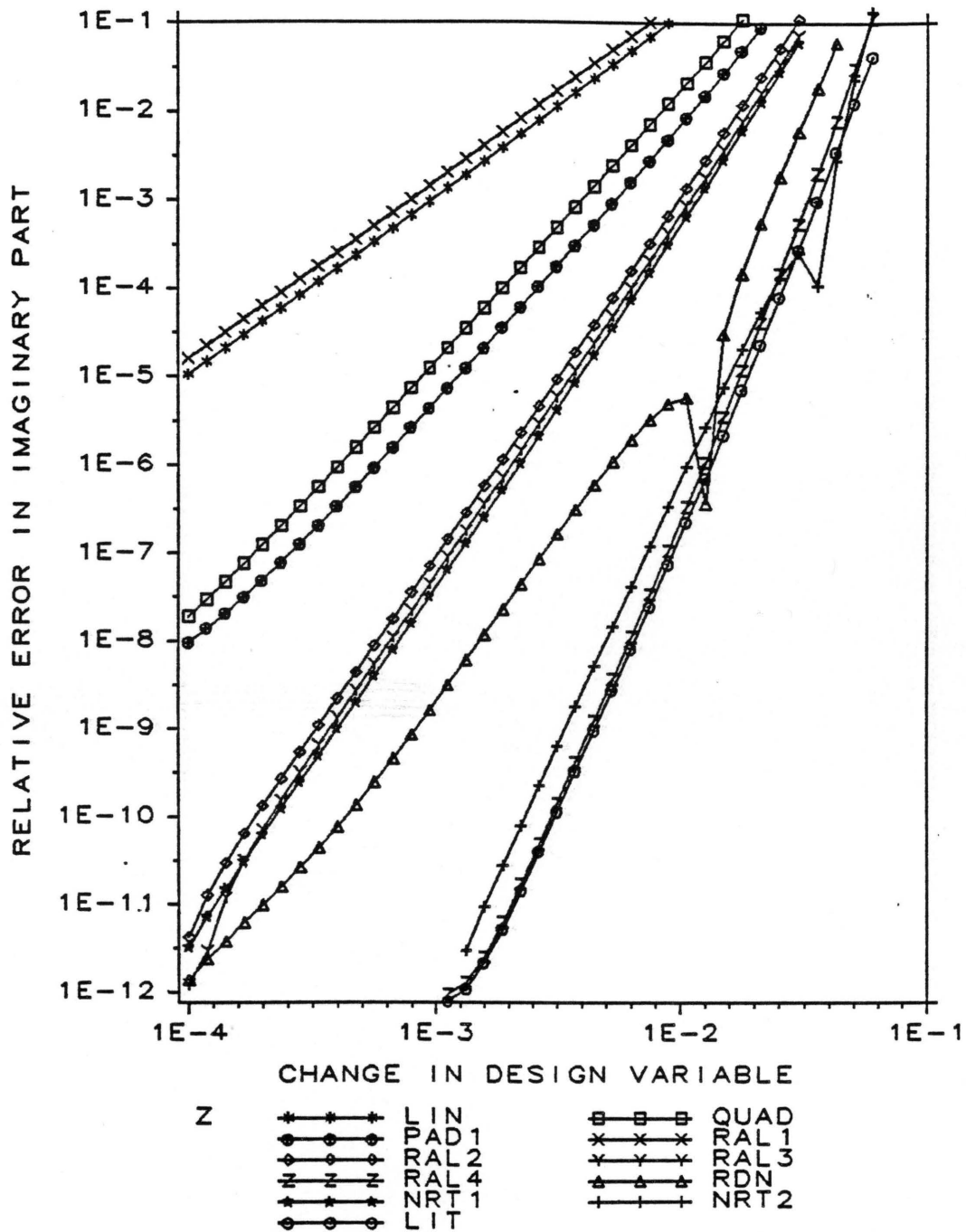


Figure 6. Relative Percentage errors in the Imaginary Part of the Eigenvalue 5 x 5 Matrix

iteration in a Rayleigh quotient. Figures 4, 5 and 6 show the errors in the absolute value, the real part and in the imaginary part of an eigenvalue of the 5 x 5 matrix plotted on a log-log scale against the change in design variable. In all the following figures, 10^i is represented in the FORTRAN notation 1E*i*. The slopes of the straight line segments in Figure 4, 5 and 6 agree very closely with the orders of the approximations derived in Section 5.2. The difference in accuracy between the high order and the low order approximations is clearly apparent.

Approximations are also applied to one of the eigenvalues of a larger matrix of order 40, generated in the same manner as the smaller matrix of order 5 before. The results obtained are shown in Figures 7, 8 and 9. The results for a flutter analysis matrix are shown in Figures 10, 11 and 12. The generation of the flutter analysis matrix is described in Section 3.2. The design variable used in the Figures 10, 11 and 12 is the reduced frequency, k defined as $\left(\frac{\omega b}{V}\right)$ where ω is the vibration frequency, b the blade semi-chord and V the air speed in far field.

The deviations from straight line behavior appearing in Figures 10, 11 and 12 at small values of Δp_α are typical instances of round-off error with which we are not concerned.

In all these Figures, approximations of equal order are indicated by straight lines of equal slope. The proportionality constant, which reflects accuracy, is indicated by the vertical position of the corresponding straight line.

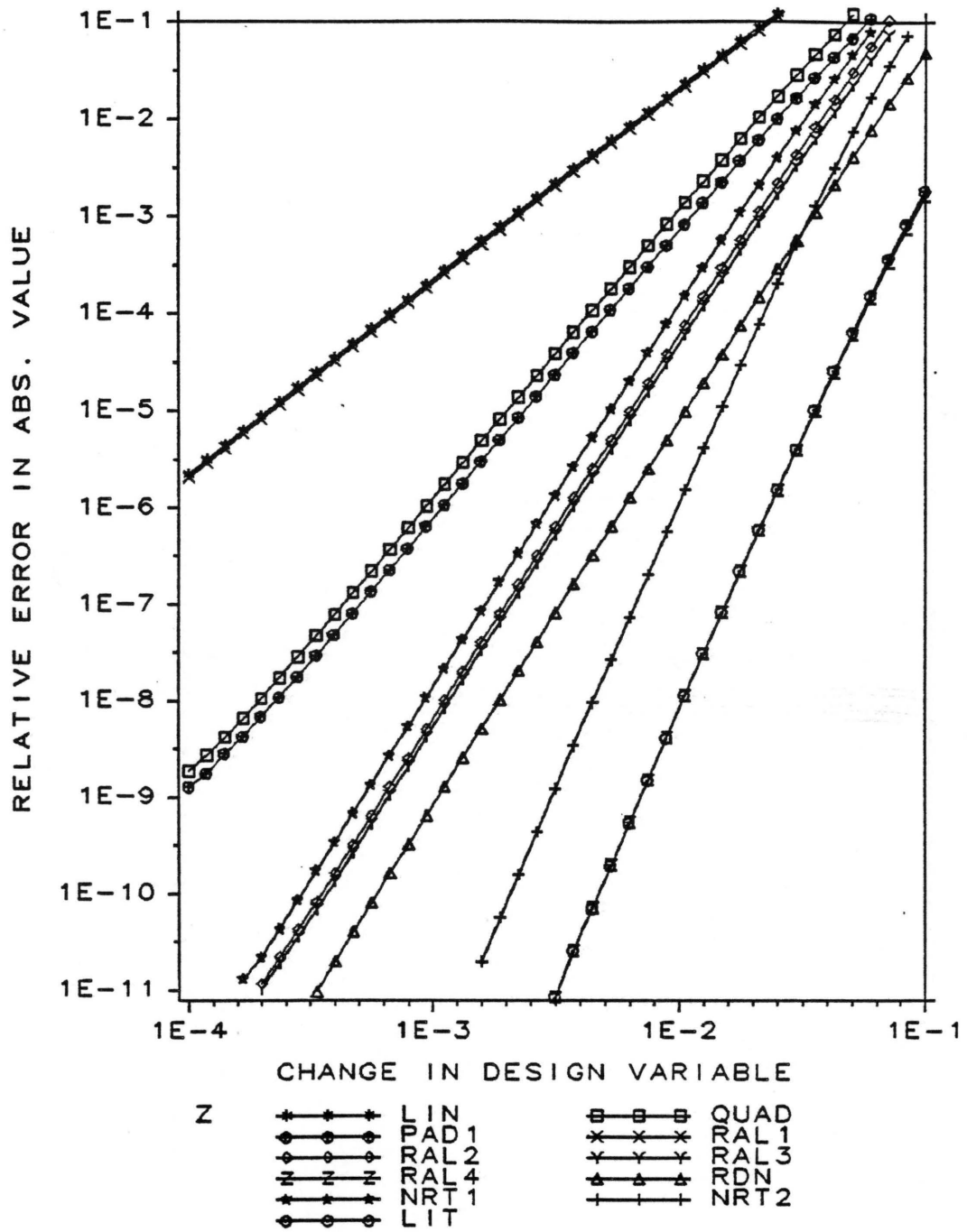


Figure 7. Relative Percentage errors in the Absolute value of the Eigenvalue 40 x 40 Matrix

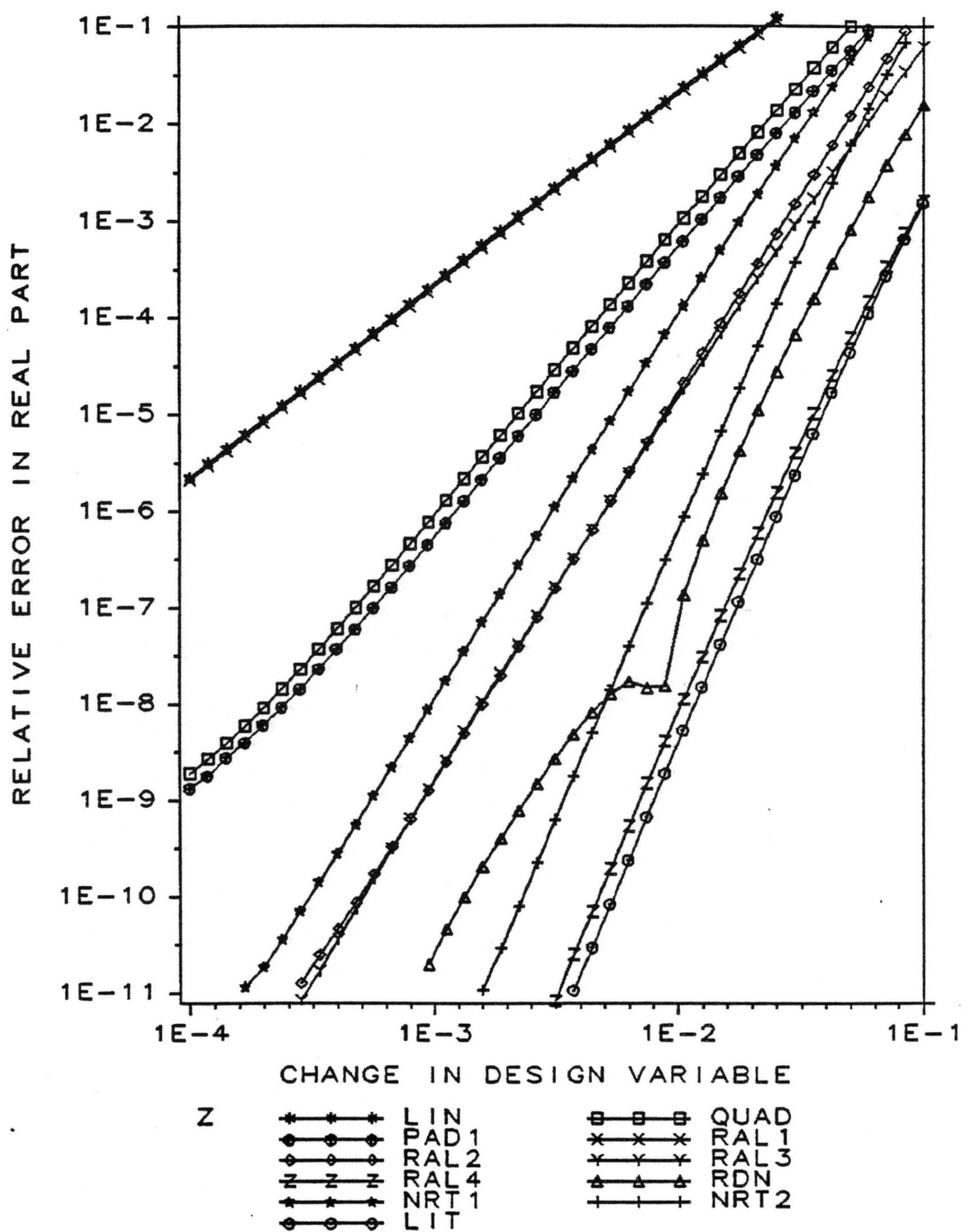


Figure 8. Relative Percentage errors in the Real Part of the Eigenvalue 40 x 40 Matrix

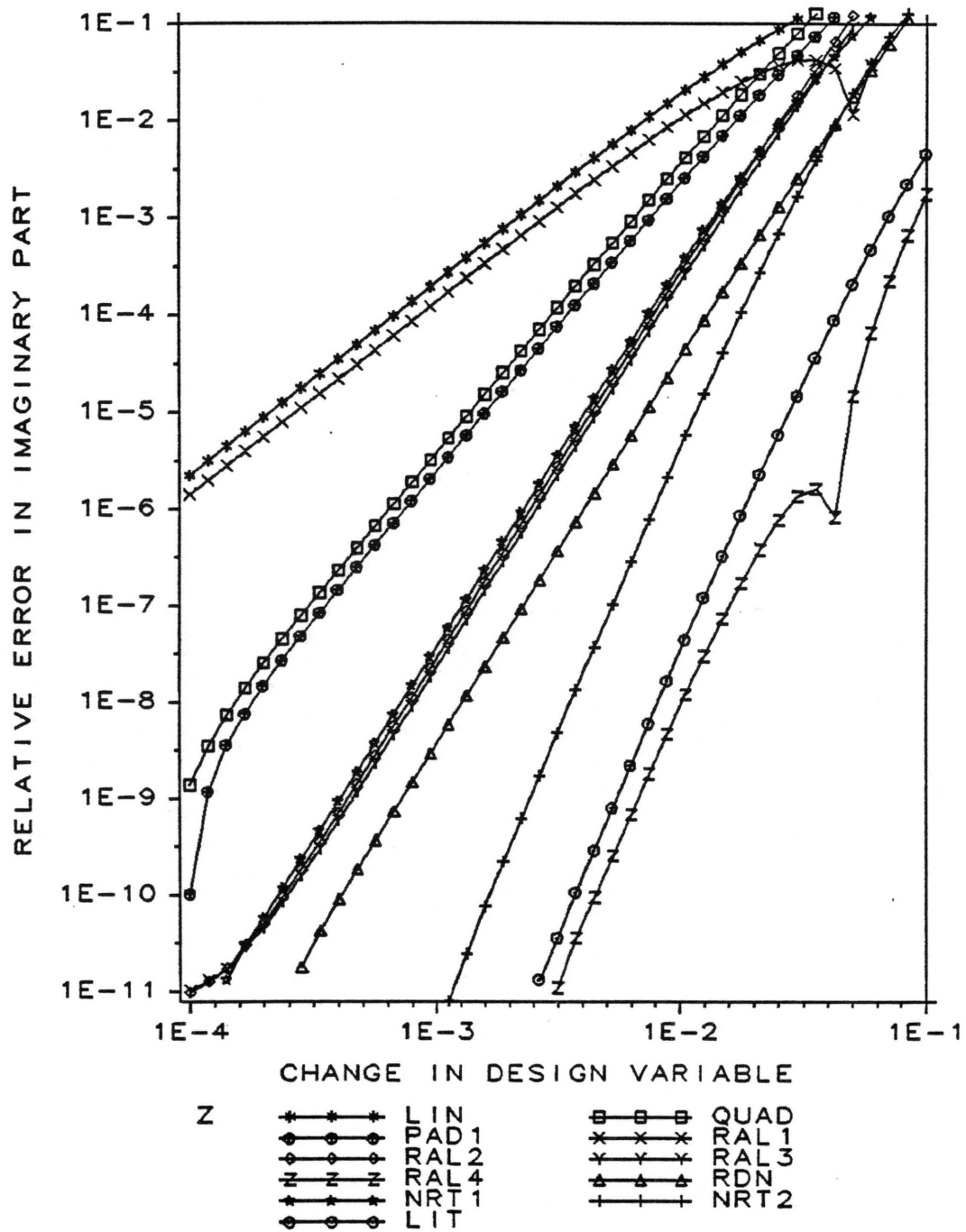


Figure 9. Relative Percentage errors in the Imaginary Part of the Eigenvalue 40 x 40 Matrix

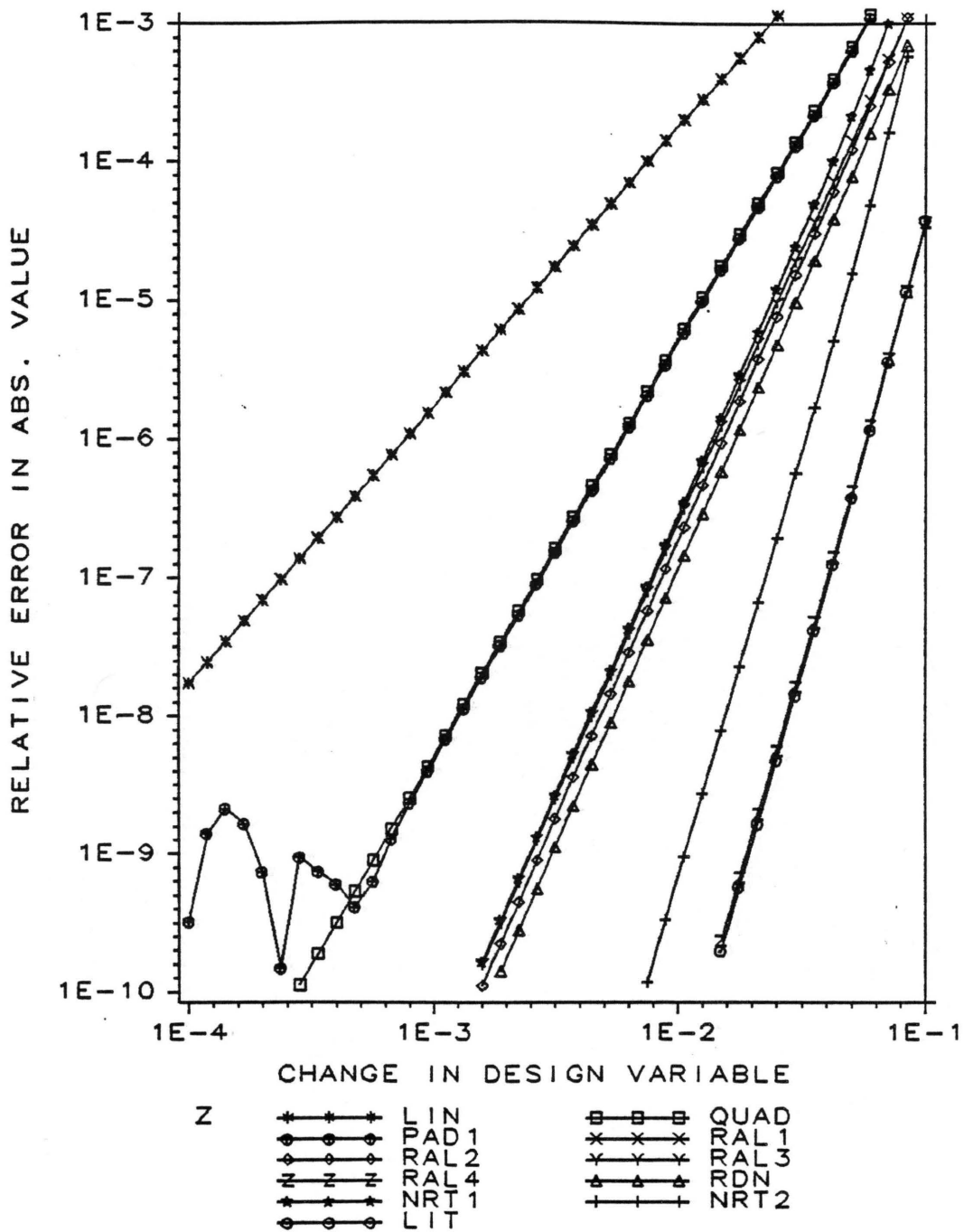


Figure 10. Relative Percentage errors in the Absolute value of the Flutter Eigenvalue 40 x 40 Matrix

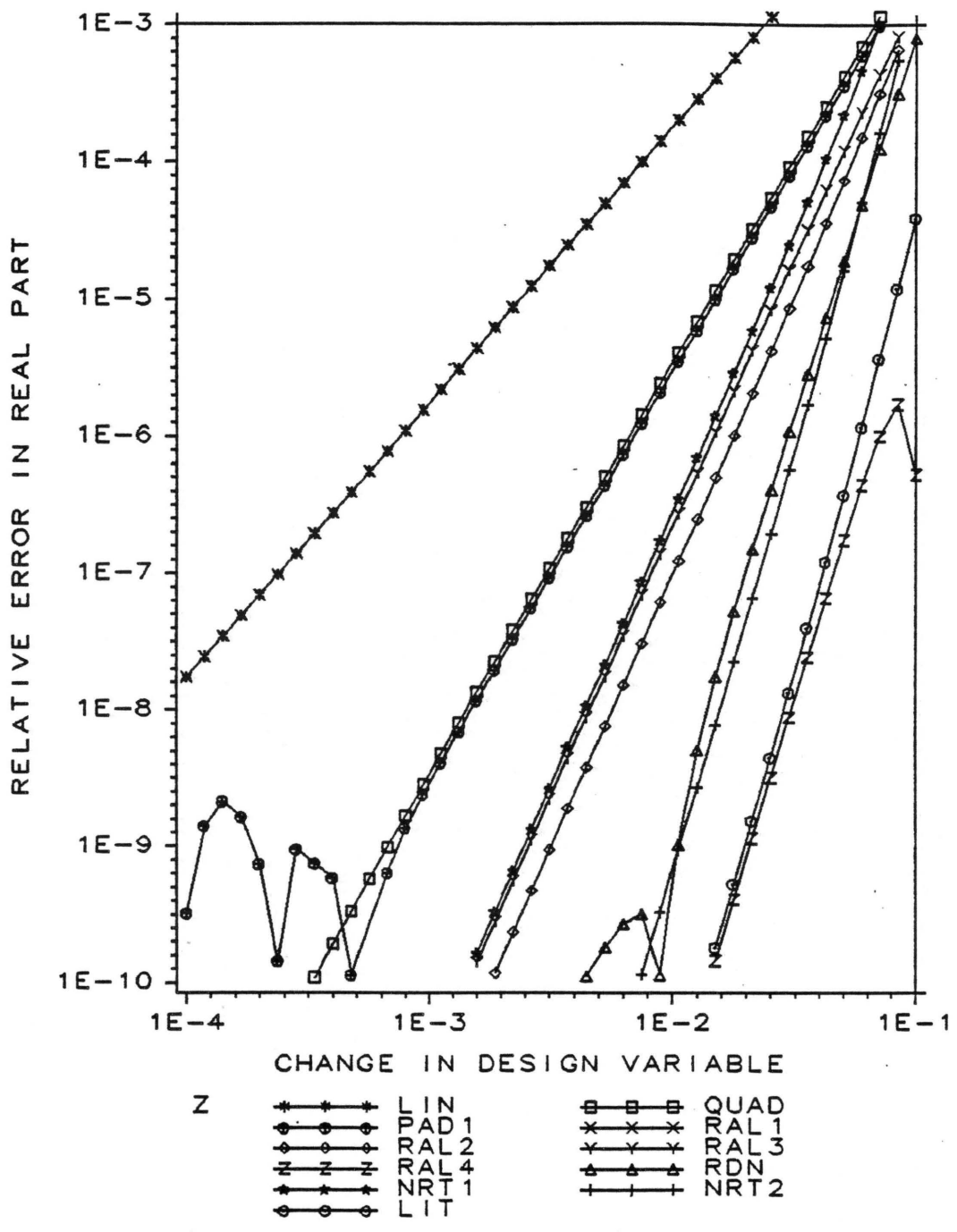


Figure 11. Relative Percentage errors in the Real Part of the Flutter Eigenvalue 40 x 40 Matrix

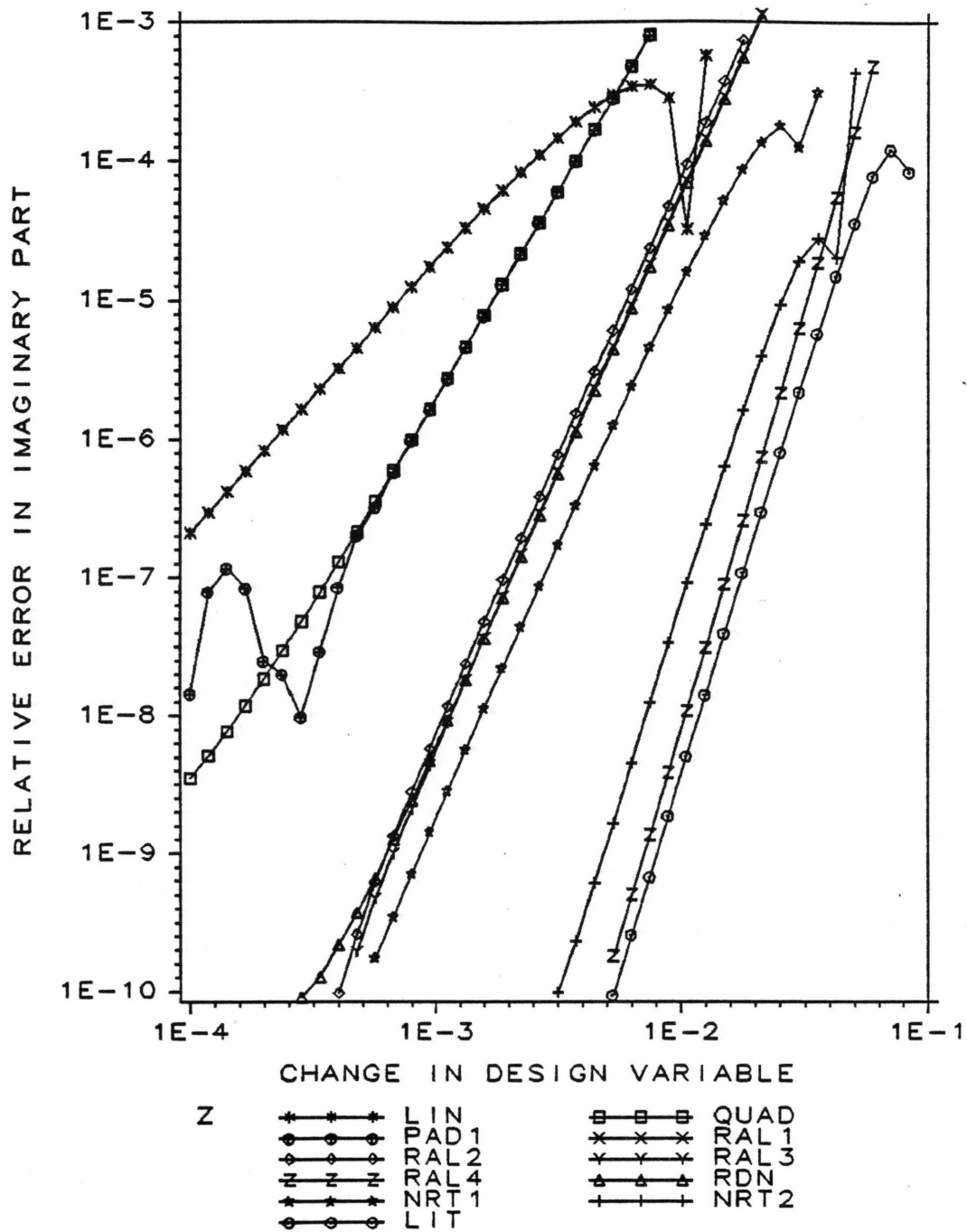


Figure 12. Relative Percentage errors in the Imaginary Part of the Flutter Eigenvalue 40 x 40 Matrix

It is seen that both the first order approximations have about the same accuracy and that the improvement achieved by the PAD1 approximation over the quadratic approximation is marginal. The reduction method approximation with good consistency shows substantially more accuracy than the other third order methods. Among the fifth order approximations, the NRT1 approximation is consistently poorer in accuracy than others.

5.3 Efficiency Considerations in Approximating

Eigenvalues

Efficiency is probably the most important consideration in the comparative evaluation of the various approximations, particularly in the context of design optimization. The comparison is once again made in terms of variables which significantly influence the cost of computing an approximation. In addition to the size of the matrix n , the number of design parameters m and the number of eigenvalues of interest l that we considered in Chapter 3, we have an additional variable in the approximation context and this is the number of design points d at which an approximation is sought based on the same nominal design.

Tables 4 and 5 present the operation counts for all the approximations studied. The operation counts include the necessary computations of the left and right eigenvectors at the nominal design represented by the matrix A_0 , if

Table 4. Operation Counts for First and Second Order Approximations

First Order Approximations	
Approximation	Operation Count
LIN	$l(\frac{7}{2}n^2 + \kappa mn^2 + dn)$
RAL1	$l(\frac{7}{2}n^2 + \kappa dn^2)$
Second Order Approximations	
Approximation	Operation Count
QUAD	$l\frac{n^3}{3} + l\binom{m}{2}n^2\kappa + lmn^2(2\kappa + 1) + dln^2$ - Direct-Adjoint Method
	$\frac{7}{2}n^3 + (\kappa + 1)mn^3 + l\binom{m}{2}n^2\kappa + dln^2$ - Adjoint Method
PAD1	Same as QUAD

Table 5. Operation Counts for Third and Higher Order Approximations

Third Order Approximations		
Approximation	Operation Count	
RDN	$l[\frac{n^3}{3} + 2(\kappa + 1)mn^2 + 2dn^2]$	Direct Method
	$\frac{7}{2}n^3 + ln^2[(\kappa + 2)m + 2d]$	Adjoint Method
RAL2	$l[\frac{n^3}{3} + 2(\kappa + 1)mn^2 + dn^2]$	Direct Method
	$\frac{7}{2}n^3 + ln^2[(\kappa + 2)m + d]$	Adjoint Method
RAL3	$dl[\frac{n^3}{3} + 2(\kappa + 1)n^2]$	Direct Method
	$\frac{7}{2}n^3 + dln^2(\kappa + 3)$	Adjoint Method
NRT1	dln^3	
Higher Order Approximations		
Approximation	Operation Count	
RAL4	$dl(\frac{n^3}{3})$	
NRT2	dln^3	
LIT	dln^3	

they are significant. But they do not include the operations for the calculation of the nominal eigenvalues λ_0 since these do not affect the comparison of the efficiency of the different approximations. When the derivatives of eigenvectors or the second derivatives of the eigenvalues are needed for an approximation, operations counts for both the Adjoint method algorithm and Direct method algorithm are given separately. The details of the Adjoint and the Direct methods are discussed in Chapters 2 and 3.

Note that the computational expense of RAL1, RAL3, RAL4 and all the fifth order approximations is independent of the number of design variables.

5.4 Discussion of Approximations

5.4.1 Case When No Derivatives are Available

The results depicted in Figures 4-12 show that both the first order approximations we studied, LIN and RAL1, are practically identical in accuracy. Comparing their operation counts, we conclude that, when first order accuracy is acceptable, LIN approximation is preferable if the number of design variables is small and the number of design points for approximation is large. When the number of design variables is large, the RAL1 approximation is more efficient.

The experience of numerical experiments also shows that the improvement achieved by the PAD1 approximation over the quadratic approximation is marginal so that we have two second order approximations of nearly the same accuracy. The behavior of the [1,1] Pade approximation based on initial approximations other than the linear and the quadratic has not been studied. However, the operation counts show that the evaluation of the second derivatives is very expensive. The third and higher order approximations are not only more accurate but are also more efficient so that the second order approximations QUAD and PAD1 can be completely dropped from consideration. Among the third order methods, RAL2, RAL3 and NRT1 have similar or same accuracy while the Reduction method RDN shows higher accuracy in most cases, sometimes close to that of some of the fifth order approximations. This is explained by the fact that the reduction method approximates the eigenvectors in a subspace of two vectors whereas RAL2 and RAL3 approximations use a subspace of only one vector. However, the reduction method is also more expensive than RAL2 and RAL3 as shown by the operation counts in Table 5.

Hence, among the third order methods, the trade-off between accuracy and efficiency determines the choice of the approximation. When accuracy is more important than efficiency, the reduction method is chosen over the others. When efficiency is more important, we have the choice between RAL2 and RAL3. The Newton-Raphson approximation(NRT1) is dropped from consideration because there are higher order methods which give higher

accuracy with the same computational expense. Note that the Direct method computation of the RAL3 approximation is more expensive than the Adjoint method computation except for very few eigenvalues of interest and very few design points of approximation. The choice between RAL2 and RAL3 is similar to that between LIN and RAL1 in the first order case. RAL2 approximation is preferable if the number of design variables is small and the number of design points for approximation is large. When the number of design variables is large, the RAL3 approximation is more efficient.

The higher order approximations are particularly efficient when the number of design variables is large and the number of design points and the number of eigenvalues of interest is small. The RAL4 algorithm is somewhat more efficient than the other two. Among these approximations, for the cases tested, the RAL4 and the LIT methods give better accuracy than NRT2. As the NRT2 approximation is no cheaper than the LIT approximation, this eliminates the NRT2 approximation from consideration. Whether there is any considerable difference in accuracy between the RAL4 and the LIT methods requires further investigation. In the absence of any such difference, the RAL4 approximation may be considered to be the best higher order approximation available in terms of accuracy and efficiency.

The computational cost of the higher order approximations escalates rapidly to equal that of the exact computation of eigenvalues. Comparing the operation count of the RAL4 approximation and the exact computation, we

note that the RAL4 approximation is more expensive than the exact computation when the product dI is above 30.

5.4.2 Case When Derivatives are Available

Most design optimization algorithms require first derivatives of constraints. Design optimization of dynamic response in structures almost always involves constraints on the eigenvalues and sometimes constraints on eigenvectors are also involved. In such cases, the first derivatives of the eigenvalues and perhaps eigenvectors are already available free for use in approximations so that the operation counts for derivative-based approximations given in Tables 4 and 5 are reduced, affecting some of the conclusions. In this section, we discuss the relative merits of the approximations when the first derivatives are already available.

We first consider the case when constraints are placed only on the eigenvalues so that the first derivatives of only eigenvalues are available free. In such a case, the operation count for only the linear approximation is affected and is shown in Table 6. The linear approximation is now an $O(n)$ process and is practically free in terms of computational expense compared to any other approximation. Thus, when constraints are placed on the derivatives of eigenvalues, the linear approximation is the best approximation unless particularly high accuracy is required. If high accuracy is desired, the conclusions of Section 5.3.1 still hold.

Table 6. Operation Counts for First Order Approximations when Eigenvalue Derivatives are Free

First Order Approximations	
Approximation	Operation Count
LIN	$1dn$
RAL1	$1(\frac{7}{2}n^2 + \kappa dn^2)$

We next consider the case when the constraints are placed on both the eigenvalues and eigenvectors so that all first derivatives are available free. In such a case, the operation counts for all derivative-based approximations are affected and are shown in Tables 7 and 8.

The derivative based approximations except the quadratic approximation are much more attractive when all the first derivatives are available. The linear approximation is again practically free so that it always makes sense to use linear approximation in terms of efficiency. The third order approximations RDN and RAL2 are now $O(n^2)$ processes and are substantially cheaper than the fifth order approximations which are still $O(n^3)$ processes. However, the RDN approximation is now twice as expensive as the RAL2 approximation and the additional computational expense of the RDN approximation is less easily justified even though it is somewhat more accurate. The quadratic approximation is again more expensive than the more accurate third order approximations. Operation counts for the fifth order approximations are not affected by the availability of the first derivatives and hence the conclusions regarding the same are also unaffected.

Table 7. Operation Counts for First and Second Order Approximations when All First Derivatives are Free

First Order Approximations	
Approximation	Operation Count
LIN	ldn
RAL1	$ld(\frac{7}{2}n^2 + \kappa n^2)$
Second Order Approximations	
Approximation	Operation Count
QUAD	$l \binom{m}{2} n^2 \kappa + lmn^2 \kappa + dln^2$ - Direct-Adjoint Method
	$(\kappa + 1)mn^3 + l \binom{m}{2} n^2 \kappa + dln^2$ - Adjoint Method
PAD1	Same as QUAD

Table 8. Operation Counts for Third and Higher Order Approximations when All First Derivatives are Free

Third Order Approximations		
Approximation	Operation Count	
RDN	$4dn^2$	
RAL2	dn^2	
RAL3	$\frac{7}{2}n^3 + dn^2(\kappa + 3)$	Adjoint Method
NRT1	dn^3	
Higher Order Approximations		
Approximation	Operation Count	
RAL4	$dI(\frac{n^3}{3})$	
NRT2	dn^3	
LIT	dn^3	

Chapter 6

Conclusions

The large computational expense associated with the flutter optimization of a cascade of rotating blades motivated this study. The problem of computational expense is attacked from two fronts, sensitivity analysis and approximations. The existing literature was surveyed in both fields and improvements are suggested. General recommendations for the selection of the most efficient algorithms are presented.

The normalization of the eigenvector needs to be properly related to its derivative. In practice, this means that the derivative of the eigenvector is to be normalized before it is used, to conform to the normalization of the eigenvector itself. When the eigenvector is not normalized in a unique manner, its derivative cannot be evaluated. It has been shown that fixing one of the components of the eigenvector is the best normalizing condition for computation of the derivative.

In the sensitivity analysis part, the algorithms presently available for computing exactly the derivatives of eigenvalues and eigenvectors are classified into Adjoint and Direct Methods. Adjoint Methods use both the left and the right eigenvectors whereas the Direct Methods use only the right eigenvectors. The Adjoint Methods and the Direct Methods found in the literature are extended to apply to eigenvectors normalized in the manner described above. Algorithms that compute approximate derivatives are not studied as their implementation is problem dependent or complicated.

The Adjoint and the Direct methods are examined for their efficiency under different sets of conditions. The choice reflects whether the solution of the adjoint problem is worth the extra computational expense. The solution of the adjoint problem is shown to be cheaper than the solution of the direct problem because left and right eigenvectors can be calculated using the same factored matrix. It is concluded that if only the first derivatives of eigenvalues are required, the solution of the adjoint problem is worth the expense since the Adjoint method is superior to the Direct Method. When first derivatives of eigenvectors are also required, the decision is dependent on the problem size, the number of design variables and the number of eigenvalues of interest. The Direct method is more competitive if the number of design variables is large and the eigenvalues of interest are few. When the first and second derivatives of eigenvalues are required, similar considerations hold. It is also shown that once the first derivatives of eigenvectors are calculated, the second

derivatives of eigenvalues are calculated more efficiently by the Adjoint method than by the Direct method.

In the approximation part, many existing approximation methods are applied to general matrices. Some new approximation methods, inspired by the computational techniques in linear algebra, are proposed. Noor's concept of global approximation vectors is simplified and then extended to general matrices to arrive at another approximation called the reduction method.

The approximations are classified as Derivative Based, Rayleigh Quotient Based, Trace Theorem Based and [1,1]Pade approximations according to their theoretical origin. In terms of accuracy, the approximations are reclassified according to the order of the error expected from the approximation. The approximation methods are also examined for computational expense as this is a significant issue in the selection of a particular method. At each order of accuracy, the approximations are compared in terms of their efficiency and general recommendations are made. Additional recommendations are made for the case when the derivatives are already available. In particular, it is concluded that the quadratic approximation is inferior to many other approximations both in accuracy and efficiency.

The analysis performed in this work is applicable also to the generalized eigenvalue problem in a straightforward manner so that all the conclusions are valid for the usual structural stability problem. However, the conclusions are limited by the assumption of distinct and well-separated eigenvalues of interest. The sensitivity analysis and approximation methods for multiple or

closely spaced eigenvalues is fraught with difficulties, numerical as well as theoretical. The multiple eigenvalue case is suggested as a topic for further research.

References

1. Collins, J. D. and Thomson, W. T., "The Eigenvalue Problem for Structural Systems with Statistical Properties", AIAA Journal, Vol. 7, No. 4, April 1969, pp.642-648.
2. Schiff, A. J. and Bogdanoff, J. L., "An Estimator for the Standard Deviation of a Natural Frequency-Parts 1 and 2 ", Journal of Applied Mechanics/Trans. of the ASME, Vol. 39, No. 2, June 1972, pp.535-544.
3. Caravani, P. and Thomson, W. T., "Frequency Response of a Dynamic System with Statistical Damping", AIAA Journal, Vol. 11, No. 2, Feb. 1973, pp.170-173.
4. Shinozuka, M. and Astill, J., "Random Eigenvalue Problems in Structural Analysis", AIAA Journal, Vol. 10, No. 4, April 1972, pp.456-462.

5. Crossley, T.R. and Porter, B., "High-order Eigenproblem Sensitivity Methods: Theory and Applications to the Design of Linear Dynamical Systems", Intl. J. of Control, Vol.10, No. 3,1969, pp.315-329.
6. Zein El-Din, H. M., Alden, R.T.H. and Chakravarti, P.C., "Second Order Eigenvalue Sensitivities Applied to Multivariable Control Systems", Proc. of the IEEE, Vol. 65, Feb. 1977, pp.277-278.
7. Zein El-Din, H.M. and Alden, R.T.H., "Second Order Eigenvalue Sensitivities Applied to Power System Dynamics", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-96, No. 6, Nov./Dec. 1977, pp.1928-1936.
8. Flax, A.H., "Comment on 'Estimation of Fundamental Frequencies of Beams and Plates with Varying Thickness'", AIAA Journal, Vol. 16, Sept.1978, pp.1022-1024.
9. Elrazaz, Z. and Sinha, N. K., "Dynamic Stability Analysis of Power Systems for Large Parameter Variations", IEEE PES Summer Meeting, Vancouver, Canada, July 15-20, 1979.
10. Elrazaz,Z. and Sinha, N. K., "Dynamic Stability Evaluation for Multimachine System: An Efficient Eigenvalue Approach", IEE Proceedings-D, Vol. 128, No. 6, Nov. 1981, pp.268-274.
11. Rudisill, C.S. and Bhatia, K.G., "Second Derivatives of the Flutter Velocity and the Optimization of Aircraft Structures", AIAA Journal, Vol.10, No.12, Dec.1972, pp.1569-1572.
12. Bhatia, K.G., "An Automated Method for Determining the Flutter Velocity and the Matched Point", J. Aircraft, Vol.11, No.1, Jan.1974, pp.21-27.

13. Haftka, R. T. and Kamat, M. P., Elements of Structural Optimization, Martinus Nijhoff Publishers, 1985.
14. Miura, H. and Schmit, L.A., "Second Order Approximation of Natural Frequency Constraints in Structural Synthesis", Intl. Journal for Numerical Methods in Engineering, Vol. 13, 1978, pp.337-351.
15. Rudisill, C.S. and Cooper, J.L., "An Automated Procedure for Determining the Flutter Velocity", J. Aircraft, Vol.10, No.7, Jul.1973, pp.442-444.
16. Junkins, J.L., Bodden, D.S. and Kamat, M.P., "An Eigenvalue Optimization Approach for Feedback Control of Flexible Structures", Proc. of SECTAM XII, May 10-11, 1984, Vol. II, pp.303-308.
17. Jacobi, C.G.J., "Uber ein leichtes Verfahren die in der Theorie der Saecular stoerungen vorkommenden Gleichungen numerisch aufzuloesen", Zeitschrift fur Reine unde Angewandte Mathematik, Vol.30, 1846, pp.51-95.
18. Wittrick, W.H., "Rates of change of Eigenvalues with reference to Buckling and Vibration Problems", J. Royal Aeronautical Society, Vol.66, 1962, pp.590-591.
19. Fox, R.L. and Kapoor, M.P., "Rates of change of Eigenvalues and Eigenvectors", AIAA Journal, Vol.6, No.12, 1968, pp.2426-2429.
20. Zarghamee, M.S., "Minimum Weight Design with Stability Constraint", J. of the Struct.Divn./ASCE, Vol.96, ST8, 1970, pp.1697-1710.
21. Fox, R.L., Optimization Methods for Engineering Design, Addison-Wesley Pub. Co. Inc., 1971.

22. Garg, S., "Derivatives of Eigensolutions for a General Matrix", AIAA Journal, Vol.11, No.8, Aug.1973, pp.1191-1194.
23. Rudisill, C.S., "Derivatives of Eigenvalues and Eigenvectors for a General Matrix", AIAA Journal, Vol.12, No.5, May 1974, pp.721-722.
24. Rudisill, C.S. and Chu, Y., "Numerical Methods for evaluating the Derivatives of Eigenvalues and Eigenvectors", AIAA Journal, Vol.13, No.6, Jun.1975, pp.834-837.
25. Cardani, C. and Mantegazza, P., "Calculation of Eigenvalue and Eigenvector Derivatives for Algebraic Flutter and Divergence Problems", AIAA Journal, Vol.17, No.4, Apr.1979, pp.408-412.
26. Taylor, D.L. and Kane, T.R., "Multiparameter Quadratic Eigenproblems", J. of Applied Mechanics/ASME, Vol.42, Jun.1975, pp.478-483.
27. Andrew, A.L., "Convergence of an Iterative Method for Derivatives of Eigensystems", J. of Comput. Physics, Vol.26, 1978, pp.107-112.
28. Crossley, T.R. and Porter, B., "Eigenvalue and Eigenvector Sensitivities in Linear Systems Theory", Intl. J. of Control, Vol.10, No.2, 1969, pp.163-170.
29. Rogers, L.C., "Derivatives of Eigenvalues and Eigenvectors", AIAA Journal, Vol.8, No. 5, May 1970, pp.943-944.
30. Plaut, R.H. and Huseyin, K., "Derivatives of Eigenvalues and Eigenvectors in Non-self-adjoint Systems", AIAA Journal, Vol. 11, No. 2, Feb.1973, pp.250-251.
31. Nelson, R.B., "Simplified Calculation of Eigenvector Derivatives", AIAA Journal, Vol. 14, No. 9, Sept.1976, pp.1201-1205.

32. Lancaster, P., "On Eigenvalues of Matrices Dependent on a Parameter", *Numerische Mathematik*, Vol. 6, No. 5, 1964, pp.377-387.
33. Pedersen, P. and Seyranian, A.P., "Sensitivity Analysis for Problems of Dynamic Stability", *International Journal of Solids and Structures*, Vol. 19, No.4, 1983, pp.315-335.
34. Reddy, D.C., "Eigenfunctions and the Solution of Sensitivity Equations", *Electronics Letters*, Vol. 4, 1968, pp.262-263.
35. Hasselman, T. K. and Hart, G. C., "Modal Analysis of Random Structural Systems", *Journal of the Engineering Mechanics Division/Proc. of the ASCE*, Vol. 98, No. EM3, June 1972, pp.561-579.
36. Adelman, H. M. and Haftka, R. T., "Sensitivity Analysis for Discrete Structural Systems-A Survey", *AIAA Journal*, Vol. 24, No. 5, May 1986, pp.823-832.
37. Morgan, B.S., "Computational Procedure for Sensitivity of an Eigenvalue", *Electronics Letters*, Vol. 2, 1966, pp.197-198.
38. Nicholson, H., "Eigenvalue and State-Transition Sensitivity of Linear Systems", *Proc. of IEE, London*, Vol. 114, 1967, pp.1991-1995.
39. Frazer, R.A., Duncan, W.J. and Collar, A.R., *Elementary Matrices*, Cambridge University Press, London, 1947.
40. Reddy, D.C., "Evaluation of the Sensitivity Coefficient of an Eigenvalue", *IEEE Trans. on Automatic Control*, AC-12, Dec. 1967, p.792.

41. Woodcock, D. L., "The Rates of Change of Eigenvalues of a Lambda Matrix and their Use in Flutter Investigations", Journal of the Royal Aeronautical Society, Vol. 70, Feb. 1966, pp.364-365.
42. Vetter, W.J., "Reply to R. G. Rains' Comments on 'An Extension to Gradient Matrices'", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-2, No. 2, 1972, p.285.
43. Nagraj, V.T., "Changes in the Vibration Characteristics due to changes in the Structure", J. Spacecraft, Vol. 7, No. 12, Dec. 1970, pp.1499-1501.
44. Paraskevopoulos, P.N., Christodoulou, M.A. and Tsakiris, M.A., "Eigenvalue Eigenvector Sensitivity Analysis of Linear Time-Invariant Singular Systems", IEEE Trans. on Automatic Control, AC-29, No. 4, Apr.1984, pp.344-346.
45. Nail, J.B., Mitchell, J.R. and McDaniel Jr.,W.L., "Determination of Pole Sensitivities by Danilevskii's Method", AIAA Journal, Vol. 15, No. 10, Oct.1977, pp.1525-1527.
46. Mitchell, J.R., Nail, J.B. and McDaniel, Jr.,W.L., "Computation of cofactors of $[sI-A]$ with Applications", IEEE Trans. on Automatic Control, AC-24, No. 4, Aug.1979,pp.637-638.
47. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965, pp.416-423,619-633,478-479,443-445.
48. Werner, W., "A Generalized Companion Matrix of a Polynomial and Some Applications", Linear Algebra and Its Applications, Vol. 55, 1983, pp.19-36.

49. Smith, B.T., et. al., "Matrix Eigensystem Routines - EISPACK Guide", Lecture Notes in Computer Science, Vol. 6, Springer-Verlag, Heidelberg, 1976, pp. 95-97.
50. Kaza, K. R. V. and Kielb, R. E., "Flutter and Response of a Mistuned Cascade in Incompressible Flow", AIAA Journal, Vol. 20, No. 8, Aug. 1982, pp.1120-1127.
51. Starnes, J. H. Jr. and Haftka, R. T., "Preliminary Design of Composite Wings for Buckling, Strength and Displacement Constraints", Journal of Aircraft, Vol. 16, No. 8, Aug. 1979, pp.564-570.
52. Prasad, B., "Explicit Constraint Approximation Forms in Structural Optimization. Part 1: Analyses and Projections", Computer Methods in Applied Mechanics and Engineering, Vol. 40, No. 1, 1983, pp. 1-26.
53. Prasad, B., "Novel Concepts for Constraint Treatments and Approximations in Efficient Structural Synthesis", AIAA Journal, Vol. 22, No. 7, July 1984, pp. 957-966.
54. Woo, T. H., "Space Frame Optimization subject to Frequency Constraints", AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference, San Antonio, TX, May 19-21, 1986, Part 1, pp.103-115.
55. Noor, A. K. and Whitworth, S. L., "Model Size Reduction for the Buckling and Vibration Analyses of Anisotropic Panels", AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference, San Antonio, TX, May 19-21, 1986, Part 2, pp.602-612.

56. Lancaster, P., *Lambda-matrices and Vibrating Systems*, Pergamon Press, 1966, pp.39-40, 82-85.
57. McCalley Jr., R. B., "Error Analysis for Eigenvalue Problems", Conference Papers-ASCE, 2nd Conference on Electronic Computation, Pittsburgh, Sept. 8-9, 1960, pp.523-550.
58. Chen, J. C. and Wada, B. K., "Criteria for Analysis-Test Correlation of Structural Dynamic Systems", *Journal of Applied Mechanics*, Trans. of the ASME, Vol. 42, June 1975, pp.471-477.
59. Chen, J. C. and Wada, B. K., "Matrix Perturbation for Structural Dynamic Analysis", *AIAA Journal*, Vol. 15, No. 8, Aug. 1977, pp.1095-1100.
60. Chen, J. C. and Garba, J. A., "Analytical Model Improvement using Modal Test Results", *AIAA Journal*, Vol. 18, No. 6, June 1980, pp. 684-690.
61. Faddeev, D. K. and Faddeeva, V. N., *Computational Methods of Linear Algebra*, W. H. Freeman and Company, 1963, pp.286-289,342-343.
62. Meirovitch, L. and Ryland, G., "A Perturbation Technique for Gyroscopic Systems with Small Internal and External Damping", *Journal of Sound and Vibration*, Vol. 100, No. 3, 1985, pp. 393-408.
63. Romstad, K. M., Hutchinson, J. R. and Runge, K.H., "Design Parameter Variation and Structural Response", *International Journal for Numerical Methods in Engineering*, Vol. 5, 1973, pp.337-349.
64. Van Ness, J. E., "Inverse Iteration Method for Finding Eigenvectors", *IEEE Trans. on Automatic Control*, Vol. AC-14, Feb. 1969, pp.63-66.

65. Paipetis, S. A. and Croustalis, M., "Approximate Determination of Eigenfrequencies in Damped Lumped-Mass Elastic Systems", *Acustica*, Vol.53, No. 6, 1983, pp.281-289.
66. Johnson, R. C., "Alternative Approach to Pade Approximants", *Pade Approximants and their Applications*, Academic Press, London, 1973, pp. 53-67.
67. Noble, B., *Numerical Methods*, Oliver and Boyd, 1964, pp. 33-36.
68. Ostrowski, A. M., "On the Convergence of the Rayleigh Quotient Iteration for the Computation of the Characteristic Roots and Vectors. III", *Archives for Rational Mechanics and Analysis*, Vol. 3, 1959, pp.325-340.

**The vita has been removed from
the scanned document**